

VŠB – Technická univerzita Ostrava
Fakulta elektroniky a informatiky

BAKALÁŘSKÁ PRÁCE

VŠB – Technická univerzita Ostrava
Fakulta elektroniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Řízení tepelné soustavy s chlazením pomocí Raspberry Pi 3 s
využitím vzdálené periferie a měniče komunikující přes Profibus DP.
Heating System Control with Cooling using the Raspberry Pi 3 using
the Remote Peripherals and Inverter Communicating via Profibus DP.

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání bakalářské práce

Student: **Josef Mudra**
Studijní program: B2649 Elektrotechnika
Studijní obor: 2612R041 Řídicí a informační systémy
Téma: Řízení tepelné soustavy s chlazením pomocí Raspberry Pi 3 s využitím vzdálené periferie a měniče komunikující přes Profibus DP
Heating System Control with Cooling using the Raspberry Pi 3 using the Remote Peripherals and Inverter Communicating via Profibus DP
Jazyk vypracování: čeština

Zásady pro vypracování:

V rámci této bakalářské práce bude řešena implementace PSD regulátoru v rámci Raspberry Pi 3 pro řízení tepelné soustavy ovládané přes vzdálenou periferii řady S7 300 od firmy Siemens. Chlazení bude realizováno pomocí 3-fázového asynchronního motoru řízeného pomocí měniče Micromaster 420 firmy Siemens přes Profibus DP.

Body zadání:

1. Sestavení modelu regulované soustavy s PSD regulátorem v prostředí Matlab/Simulink.
2. Návrh a implementace PSD regulátoru v C++ a otestování tohoto bloku v regulačním obvodu v Matlabu/Simulink.
3. Návrh a implementace PSD regulátoru z bodu 2 v rámci Raspberry Pi 3 a komunikačního rozhraní Profibus pomocí RS485.
4. Implementace řízení pomocí akčního členu - měniče Micromaster 420 v Raspberry Pi 3.
5. Vytvoření testovacího příkladu pro regulaci tepelné soustavy pomocí PSD regulátoru s chlazením v Raspberry Pi 3.
6. Zhodnocení dosažených výsledků práce.

Seznam doporučené odborné literatury:

- [1] POPP, Manfred. *The new rapid way to PROFIBUS DP: (from DP-VO to DP-V2)*. Karlsruhe: PROFIBUS, c2003.
- [2] POWELL, James a Henry VANDELINDE. *Catching the Process Fieldbus: An Introduction to Profibus for Process Automation*. New York: Momentum Press, 2013. ISBN 978-1-60650-396-6.
- [3] JONES, Clarence T. *STEP 7 Programming Made Easy in LAD, FBD, and STL - A Practical Guide to Programming S7-300/S7-400 Programmable Logic Controllers*. [s.l.], USA: Patrick-Turner Publishing, 2013. ISBN 978-1889101040.
- [4] BERGER, Hans. *Automating with SIMATIC: controllers, software, programming, data communication, operator control and process monitoring*. 5th, rev. and enl. ed. Erlangen: Publicis Publishing, 2013. ISBN 978-3-89578-387-6.
- [5] BERGER, Hans. *Automating with STEP 7 in LAD and FBD: SIMATIC S7-300/400 Programmable Controllers*. Erlangen: Publicis Publishing, 2012. ISBN 978-3895784101.
- [6] POPP, Manfred. *PROFIBUS – DP/DPV1 Basics, tips and tools for users*. Heidelberg: Hüthig, 2000. ISBN 3-7785-2781-9.
- [7] BRADBURY, Alex a Ben EVERARD. *Learning Python with Raspberry Pi*. Chichester, UK: Willey,

2014. ISBN 978-1-118-71705-9.

[8] Firemní literatura a manuály SIMATIC S7-300/400.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

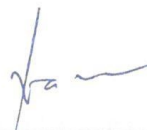
Vedoucí bakalářské práce: **Ing. Vilém Srovnal, Ph.D.**

Datum zadání: 01.09.2016

Datum odevzdání: 30.04.2018



doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry




prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlášení studenta:

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 28.3.2018


.....

Podpis

Poděkování

Rád bych poděkoval vedoucímu diplomové práce Ing. Vilému Srovnalovi Ph.D., za odbornou pomoc a konzultaci při vytváření této práce. A zvláště bych chtěl poděkovat panu doc. Ing. Jiřímu Koziorkovi Ph.D za podporu a pomocnou ruku při studiu a paní Jarmile Smějové za pomoc.

Abstrakt:

Tato práce pojednává o uceleném procesu identifikace, simulace, návrhu, ladění a implementaci regulace jak tepelné soustavy tak jakékoli jiné soustavy. Zde je popsána především identifikace a simulace v prostředí Matlab pomocí S-funkce ve které je možno přímo zkoumat chování námi naprogramovaného algoritmu. A dále pak následná implementace do mikropočítače Raspberry Pi3 a opětovná simulace pomocí spolupráce s Matlabem a řízení přes sběrnici PROFIBUS DP.

Klíčová slova:

Raspberry Pi3, PID a PSD regulace, řízení tepelné soustavy, PROFIBUS DP, návrh regulátorů, simulace HIL, MIL, PIL a SIL.

Abstract:

This work deals with a comprehensive process of identification, simulation, design, tuning and implementation of regulation of both the heat system and any other system. Here is described mainly identification and simulation in the Matlab environment using the S-function in which it is possible to directly examine the behavior of our programmed algorithm. And then the subsequent implementation into the Raspberry Pi3 microcomputer and re-simulation through collaboration with Matlab and control with PROFIBUS DP.

Key words:

Raspberry Pi, PID and PSD Controllers, drive heat system, PROFIBUS DP, design controllers. Simulation MIL, SIL, PIL, HIL

Seznam zkratek:

PID	proporcionálně, integračně, derivačně
PSD	proporcionálně, sumačně, diferenciálně
GPIO	(General Purpose Input/Output) vstupy/výstupy
CPU	centrální procesorová jednotka(unit)
MIL	model in the loop
SIL	softwarw in the loop
PIL	proces in the loop
HIL	hardwale in the loop
GCC	GNU C Compiler
USB	Universal Serial Bus
PLC	Programmable Logic Controller
u	akční zásah
y	výstup soustavy
w	zadaná teplota
e	regulační odchylka
u_{\max}	maximální akční zásah
u_{\min}	minymální akční zásah
K	zesíkení
G	přenos soustavy
T	perioda
T_i, T_d	časové konstanty

Obsah

1	Úvod.....	1
2	Teoretická část.....	2
2.1	Tepelná soustava	2
2.2	Raspberry Pi	2
2.2.1	Popis Raspberry Pi3 model B.....	3
2.3	PID regulace.....	4
2.3.1	Varianty realizace PID regulátorů.....	7
2.3.2	Číslicové regulátory	9
2.4	Modelování	10
2.4.1	Identifikace.....	10
2.4.2	Simulace.....	11
2.4.3	Model in the Loop(MIL)	11
2.4.4	Software in the loop(SIL).....	11
2.4.5	Processor in the Loop(PIL)	11
2.4.6	Hardware in the Loop(HIL)	12
2.5	Profibus teorie	12
2.5.1	Základní informace o síti PROFIBUS.....	12
2.5.2	Topologie sítě.....	13
2.5.3	Kabely pro PROFIBUS	15
2.5.4	Rozdělení sítě na segmenty	16
2.5.5	Konektory.....	17
2.5.6	Základní síť PROFIBUS	18
2.5.7	Adresování	19
2.5.8	Provoz v síti PROFIBUS.....	19
2.5.9	PROFIBUS DP slave(Decentralized Periphery)	19
3	Praktická část	26
3.1	Modelování v Matlabu	26
3.2	Simulace.....	27
3.2.1	Ladění soustavy.....	28
3.2.2	Simulace v Raspberry Pi3	30

3.3	Algoritmus PID regulátoru.....	30
3.4	Implementace	31
3.4.1	Realizace soustavy	32
3.4.2	Tepelná soustava se žárovkou a motorkem	32
3.4.3	Obvod spojitě regulace.....	35
3.4.4	Převodník k odporovým teplotním čidlům.....	35
3.4.5	Micromaster 420	36
3.5	Stavový automat.....	42
4	Závěr	46
5	Použití literatura	47

1 Úvod

Trendem dnešních dnů je zvyšování produktivity a kvality práce, potřeba hledat nové postupy šetřící čas a zároveň prostředky či lidské zdroje, což výrazně může ovlivnit cenu výsledného výrobku a tím i jeho konkurenceschopnost na trhu popřípadě ulehčení práce lidem samotným. K tomu všemu může významně přispět automatizace výrobních prostředků.

Automatizace snižuje množství chyb vytvořených lidským faktorem, pomáhá při přebírání jednotvárné činnosti či od prací nebezpečných, namáhavých nebo jen nepříjemných jako je práce v horku s kyselinami, nebezpečnými látkami. Řízení přebírají počítače a prvky umělé inteligence. Často duševní činnost třeba řízení obráběcího stroje je nahrazena číslicově řízeným obráběcím strojem. Postupně se pak vyvíjejí stroje, jež dokáží vyrábět zcela bez lidského zásahu, nebo jen za minimálního zásahu.

Řízení je neoddělitelnou součástí automatizace výroby. Kybernetika je obor, který se zabývá teorií řízení. Za jejího zakladatele se považuje matematik Norbert Wiener, který jako první zpracoval teorii zpětnovazebních systémů řízení pro účely protiletectvé obrany. Většina definic kybernetiky pochází z Wienerovy definice, který ji definoval jako „vědu o řízení a sdělování v živých tvorech a strojích.“ Základem automatizace je řízení. Řízení je cílené působení na řízený objekt tak, aby se dosáhlo určitého předepsaného cíle či stavu [5].

Potřeba regulovat nejrůznější procesy, jak v průmyslu tak v domácnosti, vedly postupně k potřebě mít k dispozici systém, který je jednak jednoduchý, všestranný, zároveň však dostatečně účinný a odolný proti vnějším vlivům. Tímto systémem je například PID regulátor a jeho diskrétní podoby, které mohou úspěšně regulovat nejrůznější fyzikální procesy, ať už jde o teplotu, tlak, otáčky, polohu, rychlost, frekvenci, chemické složení, průtok, výšku hladiny a další. Tento regulátor má velkou oblibu pro svou použitelnost v nejrůznějších aplikacích. Dobře naprogramovaný a nastavený PID regulátor stále zůstává v praxi daleko nejčastěji používaným regulátorem při relativně jednoduchém nastavování[1]. Mnohé firmy vyrábějí PID regulátory v podobě bloku(např. Eurotern, Autonics, R&B, ...). Tyto bloky umožňují automatické naladění regulátoru, nastavení různých termočlánků, regulace teploty respektive chlazení, volit různé režimy regulace atd. Uvádí se, že dokonce 95% regulačních algoritmů je typu PID a že velká většina z nich navíc využívá pouze proporcionální a integrační složku[6].

Cílem této práce je projít základní kroky návrhu a implementace algoritmu PID regulátoru a s pomocí mikrokontroleru Raspberry Pi3 řídit tepelnou soustavu, složenou ze žárovky, kovové destičky a ventilátoru.

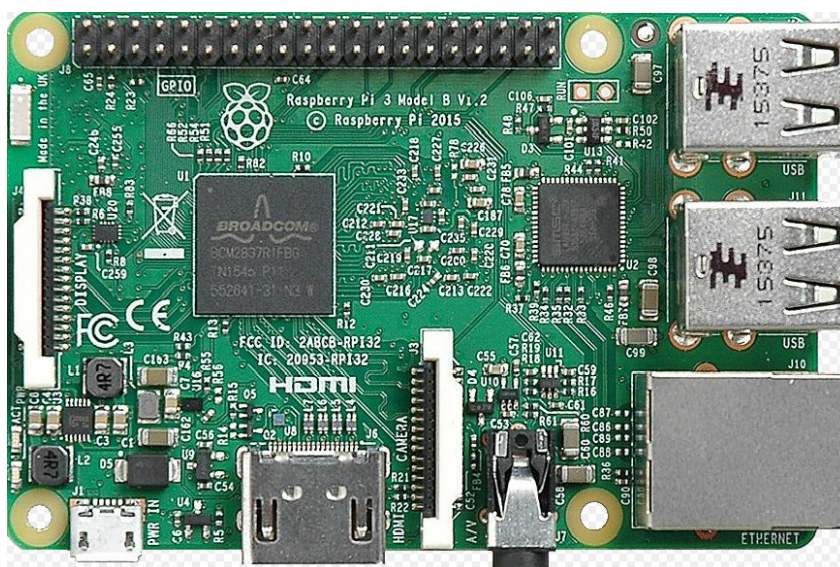
Celý proces návrhu regulátoru může být započat měřením odezvy soustavy a její následná identifikace čímž získáme model soustavy. Simulace a ladění v modelovacím prostředí Matlab nám pomůže správně naladit soustavu, což je důležité především v případech, kdy může dojít k poškození soustavy či součástí soustavy, což není náš případ. A následná aplikace v daném kontroléru, kterým je v tomto případě Raspberry Pi3. Jde o jednodeskový minipočítač o velmi malých rozměrech a poměrně nízké ceně, jehož úkolem je sloužit především pro účely výuky a vzdělání. Jak jeho spoluautor prohlásil cena bude menší než cena učebnice.

2 Teoretická část

2.1 Tepelná soustava

Naše tepelná soustava se nachází v laboratořích VŠB TO fakulty elektroniky a informatiky. Skládá se s hliníkové destičky, která je ozařována žárovkou pro ohřev a ventilátorem je ochlazována. Třífázový motor ventilátoru je řízen modulem Mikromáster 420, žárovka je řízena PLC modulem ET200M pomocí přípravku pro spojitou regulaci napětí a převod odporových čidel na napětí. Řídící algoritmus je na mikrokontroleru Raspberry Pi3 (obrázek 1.), který komunikuje s moduly přes USB/RS485 a sběrnici PROFIBUS DP.

2.2 Raspberry Pi



Obrázek 1. Raspberry Pi 3 Model B

Počítač Raspberry Pi je dostatečně levný, aby si jej děti mohly koupit za několikaměsíční kapesné, všechno příslušenství, které ke své činnosti potřebuje, již naleznete doma: televizor, SD kartu můžete použít ze starého fotoaparátu či telefonu, nabíječku telefonů, klávesnici, myš. Je dostatečně malý, aby jste si jej mohli vzít do kapsy a přenesli kamkoli potřebujete. Když se něco pokazí, nic se neděje, stačí vyměnit SD kartu a vše funguje jako po zakoupení. Chcete-li se vydat na dlouhou cestu, na jejímž konci dokážete svůj počítač Raspberry Pi programovat, máte všechny potřebné nástroje, funkce a výukové materiály k dispozici hned poté, co počítač zapnete[4].

V roce 2012 byl vyvinut britskou nadací Raspberry Pi Foundation s cílem podpořit zájem dětí o informatiku, výuku informatiky ve školách a seznámit studenty s tím, jak mohou počítače řídit různá zařízení.

Na začátku první pokusy nepřipomínaly dnešní Raspberry Pi, kdy spoluautor Eben Upton pájel první model s procesorem Atmel, kterým dokázal ovládat televizi. Tyto prototypy s 512 kB RAM a několika Mips výpočetního výkonu se svými výkony velmi podobaly osmibitovým mikropočítačům. Firma Broadcom, výrobce polovodičů, dodala levný, ovšem vysoce výkonný hardware, který tato společnost vyrábí pro špičkové mobilní telefony, vhodné pro HD video či fotografování ve vysokém rozlišení.

Tyto čipy poskytují funkce univerzálního výpočetního zpracování, 3D grafiky, videa a paměti integrované do jediného pouzdra BGA velikosti nehtu. Tyto mikročipy mají velmi nízkou spotřebu, ale nabízejí značné možnosti zejména v oblasti multimediální. Koncem roku 2011 byl na světě další prototyp a zájem byl obrovský. Výrobu převzala komerční firma element14 a RS Components. První objednávky chodily 7x za sekundu. Objednávky za první tři měsíce dosáhly půl milionu kusů. Dnešní Raspberry Pi má velkou komunitu, která vám dokáže pomoci se všemi nesnázemi. Na webových stránkách <http://www.raspberrypi.org> zveřejňují každý den nějakou novou zajímavost. Hlavní myšlenka ovšem je vyrobit počítač v ceně učebnice, na kterém by bylo vše potřebné pro výuku programování a ovládání vlastního hardwaru a to vše především pro děti.

2.2.1 Popis Raspberry Pi3 model B

V únoru 2016 byl uveden do prodeje model Raspberry Pi 3 model B, který je poprvé vybaven 64bitovým CPU o taktu 1,2 GHz. Jde o čtyř jádro ARM Cortex-A53 a dle výrobce je o 50 % rychlejší než CPU v Raspberry Pi 2. Raspberry Pi 3 je třetí generace Raspberry Pi. Jde o malý jednodeskový počítač o velikosti platební karty. Primárním operačním systémem je Raspbian. Označení „Raspberry Pi“ je registrovanou ochrannou známkou. Použitý mikroprocesor je z rodiny ARM, takže je srovnatelný s běžným smartphonem[3]. Architektura ARM využívá redukované instrukční sady RISC, což se významně podílí na nízké spotřebě energie.

Pin#	NAME	Connection		Connection	NAME	Pin#
01	3.3V			5V (Cupcade)	5V	02
03	GPIO 2			5V (Powerboost)	5V	04
05	GPIO 3			GND (Powerboost)	Ground	06
07	GPIO 4	START			GPIO 14	08
09	Ground	GND (Cupcade)			GPIO 16	10
11	GPIO 17	UP		SELECT	GPIO 18	12
13	GPIO 27	DOWN		GND (Select/Start)	Ground	14
15	GPIO 22	LEFT		RIGHT	GPIO 23	16
17	3.3V			A	GPIO 24	18
19	GPIO 10	B		GND (ABXYR)	Ground	20
21	GPIO 09	X		Y	GPIO 25	22
23	GPIO 11	L Shoulder		R Shoulder	GPIO 08	24
25	Ground	GND (L)			GPIO 07	26
27	ID_SD				ID_SC	28
29	GPIO 05				Ground	30
31	GPIO 06				GPIO 12	32
33	GPIO 13				Ground	34
35	GPIO 19				GPIO 16	36
37	GPIO 26				GPIO 20	38
39	Ground				GPIO 21	40

Obrázek 2. Popis pinů GPIO

Parametry:

Datum vydání	Únor 2016
Architektura	ARMv8-A (64/32-bit)
SoC	Broadcom BCM2837
Procesor (CPU)	1.2 GHz 64-bit quad-core ARM Cortex-A53
Video (GPU)	Broadcom VideoCore IV @ 250 MHz (BCM2837: 3D part of GPU @ 300 MHz, video part of GPU @ 400 MHz) OpenGL ES 2.0 (BCM2835, BCM2836: 24 GFLOPS / BCM2837: 28.8 GFLOPS) MPEG-2 and VC-1 (with license), 1080p30 H.264/MPEG-4 AVC high-profile decoder and encoder (BCM2837: 1080p60)
Paměť (SDRAM)	1 GB (sdílená s GPU)
USB 2.0 porty	4 (přes zabudovaný pětiportový USB hub; jeden USB port vnitřně propojen s ethernet portem)
Video vstup	15-pinový MIPI konektor kamerového rozhraní (CSI) HDMI (rev 1.3 & 1.4), 14 HDMI rozlišení od 640×350 do 1920×1200
Video výstup	plus různé PAL a NTSC standardy, kompozitní video (PAL a NTSC) via 3,5 mm TRRS jack sdílený s výstupem zvuku, MIPI konektor rozhraní displeje (DSI)
Zvukový vstup	Přes I2C rozhraní
Zvukový výstup	Analogový (přes 3,5mm jack), digitální (přes HDMI), I2S
Interní paměť	MicroSDHC, USB Boot Mode
Integrovaná síť	10/100 Mbit/s Ethernet + WiFi 802.11n a Bluetooth 4.1
Nízkoúrovňové periferie	17× GPIO plus stejné funkce a HAT ID sběrnice
Jmenovitý výkon	300 mA (1,5 W) v průměru na volnoběh 1,34 A (6,7 W) při maximálním zatížení (připojený monitor, klávesnice, myš a WiFi)
Rozměry	85,60 mm × 56,5 mm × 17
Hmotnost	45 g

Pro zavedení systému a trvalé uchování dat je určena MikroSD.

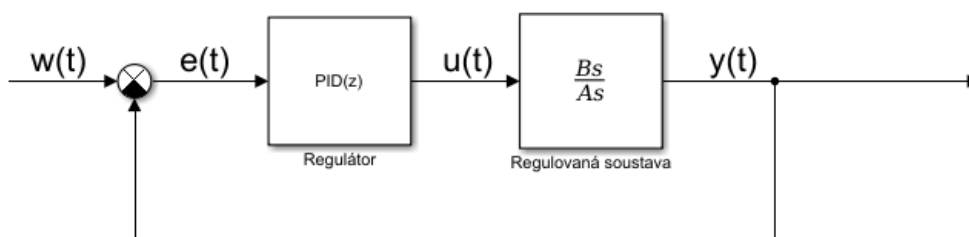
2.3 PID regulace

Za počátky éry PID regulátorů lze považovat období 1915-1940, kdy vznikaly proslulé firmy Bristol, Fisher, Foxboro, Honeywell, Leeds & Nortrup a Taylor Instrument. Proporcionálně integrační regulátory byly použity v průmyslu již mnohem dříve. Proporcionální zpětná vazba je součástí dobře známého odstředivého regulátoru vynalezeného kolem roku 1750, který byl užit pro řízení otáček větrného mlýnu. Podobný řídil otáčky Wattova parního stroje (1788). Regulátor s derivační složkou byl poprvé sestaven ve společnosti Taylor Instrument (1935). Od té doby zůstávají základní funkční vlastnosti průmyslového regulátoru v podstatě beze změn.[6]

Regulace se uskutečňuje prostřednictvím regulačního obvodu. Regulovaná soustava je předmětem regulace. Regulátor je zařízení, které uskutečňuje regulaci a je za tímto účelem úmyslně sestrojeno[5]. Regulovaná soustava je v našem případě hliníková destička ozařovaná žárovkou, regulovanou veličinou je teplota destičky, akční zásah v našem případě výstup z PLC modulu. Výstupní teplotu označujeme jako výstup soustavy y . Žádaná hodnota je veličina w , pomocí níž nastavujeme hodnotu, které chceme dosáhnout. Hodnota regulované veličiny se v obvodu neustále měří a porovnává se zadanou hodnotou, čímž získáme regulační odchylku e (1.). Právě tu odstraňuje regulátor svým akčním zásahem. Díky záporné zpětné vazbě je akční zásah takového charakteru, aby se zmenšila regulační odchylka. Jeli zadání rovno regulované veličině a regulační odchylka nulová pak regulátor neprovádí žádnou změnu.

$$e = w - y \quad (1.)$$

Jakmile je regulační odchylka nenulová, provádí se akční zásah u . Všechny tyto veličiny probíhají v čase $y(t), w(t), u(t), e(t)$. Základní blokové schéma obvodu je na Obrázku 3.



Obrázek 3. Zpětnovazební řídicí systém s PID regulátorem

Lineární spojitý systém se vstupem $u(t)$ a výstupem $y(t)$ je obecně popsán diferenciální rovnicí, přenos systému. Řád nejvyšší derivace vstupní veličiny je vždy vyšší nebo roven řádu nejvyšší derivace výstupní veličiny. Tato rovnice nám umožňuje určit odezvu či simulovat chování systému. Pro tento případ je přenos systému nejčastěji používaným způsobem popisu lineárních regulačních systémů. Je definován jako poměr obrazu výstupní ku vstupní veličiny vše v Laplaceově transformaci. Při nulových počátečních podmínkách[5].

Klasické rozdělení regulátorů je na direktní a indirektní. Tedy na přímé a nepřímé. Direktní regulátory jsou regulátory, které nepotřebují k regulaci vnější energii. Mezi nejznámější patří Wattův regulátor otáček s roztěžníkem používaný dříve u parních strojů. V dnešní době se direktní regulátory již téměř nepoužívají, jejich dynamické vlastnosti nebyly dobré.

Indirektní regulátory používají vždy vnější pomocný zdroj energie. Podle této energie se dělí na regulátory elektrické, pneumatické a hydraulické. Nejpoužívanější jsou eklektické regulátory. Většinou jsou to elektrická zařízení např. operační zesilovače, pouze akční členy mohou být elektromechanické. Výhodou jsou dobré regulační vlastnosti, rychlá odezva, vysoká účinnost, čistý a bezhlučný provoz. Dnes nemají konkurenci v jiných typech regulátorů[5].

Podle průběhu vstupní a výstupní veličiny se dělí na spojité a nespojité(diskrétní). Spojité regulátory pracují se spojitým signálem a jeho základním stavebním kamenem jsou operační zesilovače.

Regulace je poměrně dobrá a návrh je poměrně snadný. V dnešní době se využívají regulátory diskrétní, které jsou řízeny mikroprocesorem.

Regulátor není jeden člen, ale jeho základní části jsou měřicí člen, ústředí člen a akční člen zapojené do série. Měřicí člen zjišťuje skutečnou hodnotu řízené veličiny, převádí ji na el. veličinu a pak pomocí převodníku ji převede na informaci. Tuto hodnotu předává ústřednímu členu, který vypočítá akční zásah. Ústřední člen má rozhodující vliv na regulační pochod.

PID respektive PSD regulátor je zpětnovazební řídicí systém s jednou proměnou. Působení zpětné vazby má zásadní vliv na řadu parametrů, jako je stabilita, šířka pásma, celkové zesílení, citlivost[2].

Při realizaci regulátoru se používají tři základní složky a to proporcionální, integrační a derivační. Následně s těmito složkami lze složit několik základních typů regulátorů a to P, I, PD, PI a PID(případně S, PS, PSD). Pro regulování tepelné soustavy je nejvhodnější použití PID regulátoru.

- Proporcionální regulátor

$$u = \begin{cases} u_{\max} & \text{je-li } K_e + u_b > u_{\max} \\ K_e + u_b & \text{je-li } K_e + u_b \in \langle u_{\min}, u_{\max} \rangle \\ u_{\min} & \text{je-li } K_e + u_b < u_{\min} \end{cases} \quad (2.)$$

Kde K je zesílení regulátoru, u_b je obvykle voleno jako střed $((u_{\max} + u_{\min})/2)$ v rozsahu akční veličiny nebo je zadávána ručně (manulal reset). Uvažujme regulační smyčku obrázek 3. s proporcionálním regulátorem(2) a statickým systémem, jehož statické zesílení je K_0 , d je libovolná konstantní porucha, potom pro ustálený stav pak platí:

$$e = \frac{1}{1 + KK_0} w - \frac{K_0}{1 + KK_0} (u_b + d) \quad (3.)$$

Vhodnou volbou u_b je možné docílit nulovou regulační odchylku. Nevýhodou je zesílení K je shora omezeno požadavkem na stabilitu uzavřené smyčky[6][8].

- PI regulátor

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right] \quad (4.)$$

K je zesílení, T_i je integrační časová konstanta regulátoru. Díky integrátoru se docílí, že regulační odchylka se rovná nule v ustáleném stavu. Kdyby tomu tak nebylo, docházelo by díky integraci nenulové odchylky ke změně výstupu u , a smyčka by nebyla v ustáleném stavu[6].

- PID regulátor – snaha zlepšit stabilitu uzavřené smyčky s PI regulátorem vedla k zavedení derivační složky.

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (5.)$$

T_d je derivační časová konstanta. Derivační složka umožňuje předvídat budoucí chování procesu. Velkým omezením derivační složky je v případech, kdy systém má velké dopravní zpoždění nebo regulovaná veličina je zatížena velkým rušením měření[6][8].

2.3.1 Varianty realizace PID regulátorů

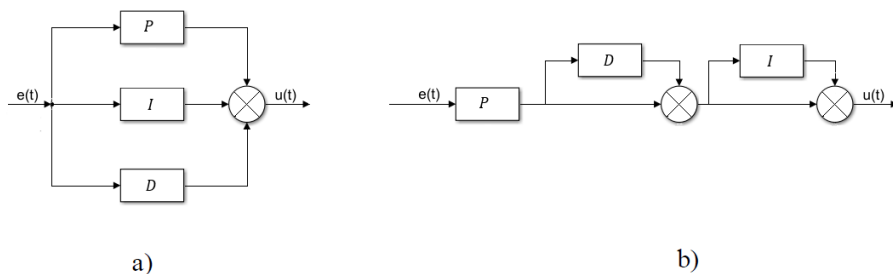
- Paralelní a sériová realizace

Ideální PID regulátor je popsán přenosem

$$G(s) = K \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (6.)$$

Velká většina výrobců používá

$$G'(s) = K' \left(1 + \frac{1}{T_i' s} \right) (1 + T_d' s) \quad (7.)$$



Obrázek 4.a) paralelní b) sériová realizace PID regulátoru

Na obrázku 4. a) standardní zapojení, často nazývané paralelní nebo též neinteraktivní přenosu (6.), poněvadž všechny tři složky se dají nastavit nezávisle, na obrázku 4.b) jsou všechny tři složky závislé(7). Neinteraktivní je obecnější než interaktivní[6][8].

- Filtrace derivační složky

Jestliže je regulovaná veličina zatížena vysokofrekvenčním šumem, potom derivační složka může způsobit nežádoucí kmitání akční veličiny. Vyplývá to z následující úvahy. Sinusový šum:

$$n = a \sin \omega t \quad (8.)$$

Způsobí v derivační složce

$$u_n = k T_d \frac{dn}{dt} = a k T_d \omega \cos \omega t \quad (9.)$$

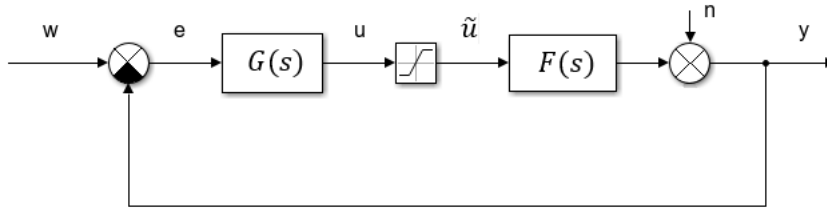
Odkud plyne, že jestliže má šum dostatečně velkou frekvenci ω , potom amplituda akční veličiny může mít libovolně velkou amplitudu. Tento problém vyřešíme zařazením filtru 1. řádu.

Derivační složka má potom přenos:

$$\frac{kT_d s}{\frac{T_d}{N}s + 1} \quad (10.)$$

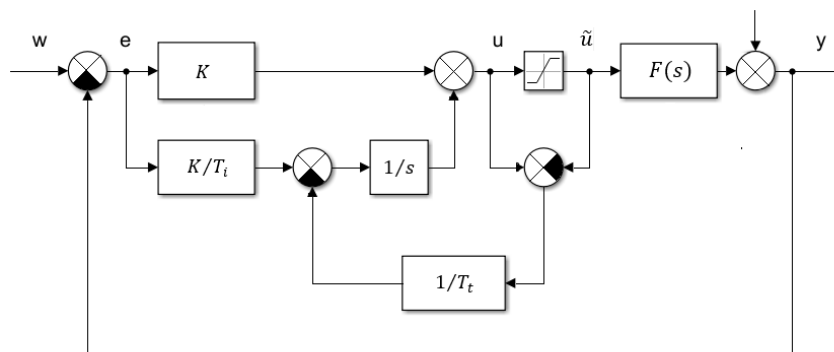
Kde N je parametr určující časovou konstantu T_d/N filtru. Typické hodnoty N leží v intervalu $\langle 3, 20 \rangle$. Pro malé hodnoty N je derivační složka málo účinná[6][8].

- Unášení integrační složky(Integrátor Windup)



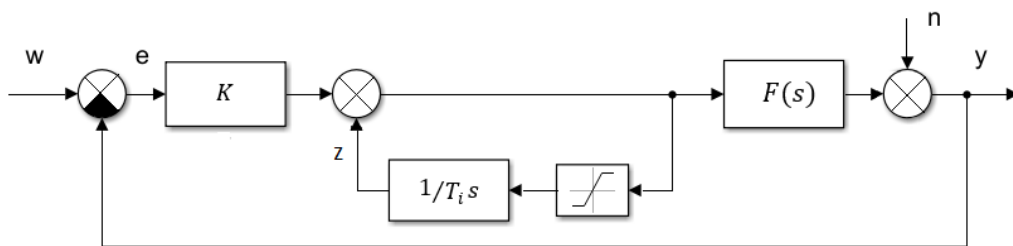
Obrázek 5. Regulační smyčka s respektováním saturace vstupu řízeného procesu

Každý řídicí člen je schopen realizovat akční zásah jen v určitém rozmezí. Z tohoto důvodu je nutné doplnit model o blok saturace. Z obrázku 5. je zřejmé, že skutečný vstup procesu \tilde{u} může být nyní různý od výstupu regulátoru u . V takovém případě je však regulační smyčka rozpojena a dochází k nežádoucímu unášení stavu regulátoru. V důsledku toho vznikají dlouhé nežádoucí přechodové jevy po návratu do lineárního režimu. Předpokládejme, že regulační odchylka e je z nějakého důvodu po dostatečně dlouhou dobu kladná, potom se realizovaný řídicí signál \tilde{u} dostane do saturace na hodnotě u_{max} . Jestliže kladná odchylka trvá, potom integrační složka dále roste a rozdíl mezi u a \tilde{u} stoupá. Tento stav je ukončen teprve poté, co se změní znaménko odchylky. Nyní musí být zase odchylka dostatečně dlouho záporná, aby došlo k od integrování integrační složky, aby se regulační smyčka vrátila do lineárního režimu, kde platí $u = \tilde{u}$. Klasický způsob, jak odstranit unášení integrační složky u PI regulátoru, je naznačen na obrázku 6.[6][8].



Obrázek 6. Klasická struktura PI regulátoru zabráňující unášení integrační složky (anti-reset windup)

Jakmile je realizovaná akční veličina \tilde{u} v saturaci, potom je zpětnovazební signál $u - \tilde{u}$ kladný a působí na stav integrátoru tak, aby byla obnovena rovnost $u = \tilde{u}$ rychlost konvergence $u \rightarrow \tilde{u}$ je dána časovou konstantou T_t . Tedy integrační složka je přepočítána tak, že výstup regulátoru u je přesně roven horní saturační mezi $max. u$ a tím je odstraněno unášení integrační složky. Jiný způsob původně používaný u interaktivních analogových regulátorů je uveden na obrázku 7.



Obrázek 7. Jiný způsob odstranění unášení integrační složky u PI regulátoru

V případě, že vnitřní smyčka pracuje v lineárním režimu, je její přenos dán vztahem

$$\frac{1}{1 - \frac{1}{T_i s + 1}} = \frac{T_i s + 1}{T_i s} \quad (11.)$$

a tedy regulátor realizuje standardní PI algoritmus. Jestliže ve vnitřní smyčce dochází k saturaci, potom je zřejmě též signál z omezen a tak není možné, aby docházelo k unášení integrační složky[6]. Variant PID regulátorů je samozřejmě mnohem více, každý výrobce si upravuje algoritmus dle svého. Základ je však stejný.

2.3.2 Číslicové regulátory

Od číslicového regulátoru očekáváme totožnou funkci jako u regulátoru spojitého. To znamená zesilovat, integrovat a derivovat regulační odchylku. Proto při sestavování diskrétního algoritmu vycházíme ze spojitého regulátoru PID. PID regulátor je popsán rovnicí (5.). Číslicovou verzi získáme diskretizací derivace a integrace. Spojitý signál nahradíme tzv. stupňovitou náhradou zleva, obdélníky zleva. Hodnotu integrálu pak určíme sečtením plochy těchto obdélníků.

$$\int_0^{kT} e(t) dt = T \sum_{i=1}^k e(i) \quad (12.)$$

Derivaci nahradíme diferencemi podle rovnice (13.).

$$\frac{de}{dt} = \frac{e(k) - e(k-1)}{T} \quad (13.)$$

Ve jmenovateli zatím necháme vzorkovací periodu T . Po dosazení do rovnice spojitého regulátoru PID a dosazení diskrétního času dostaneme rovnici:

$$u(t) = K \left[e(k) + \frac{T}{T_i} \sum_{i=1}^k e(i) + \frac{T_d}{T} [e(k) - e(k-1)] \right] \quad (14.)$$

Jelikož se integrační složka získá sumací a derivační složka zpětnou diferencí, nazývá se tento regulátor proporcionálně-sumačně-diferenční neboli PSD.

2.4 Modelování

2.4.1 Identifikace

U lineárních regulovaných soustav se nejčastěji používají matematické modely ve tvaru přenosů s vyjádřenými konstantami. Získání matematického modelu se nazývá identifikace. Identifikace může být analytická nebo experimentální. Každá regulovaná soustava vyžaduje jiný způsob identifikace. Jestliže na vstup systému přivedeme vstupní funkci jednotkového skoku

$$u = \begin{cases} 1 & \text{pro } t \geq 0 \\ 0 & \text{pro } t < 0 \end{cases} \quad (15.)$$

Potom na výstupu obdržíme přechodovou charakteristiku g danou vztahem

$$g(t) \triangleq y(t) = \int_0^t h(\tau) d\tau \quad (16.)$$

Předpokládá se že identifikovaná soustava je vhodně upravena (filtrace, vyhlazení) a že se pracuje s přírůstkovými veličinami, to je, že vycházejí z počátku souřadnic. Jeli přechodová charakteristika nekmitavá, proporcionální, pak nejjednodušší identifikační metoda je určení doby průtahu T_u a doby ustálení T_n které vytkne tečna v inflexním bodě na časové ose a ustálené hodnotě (obrázek 8.)

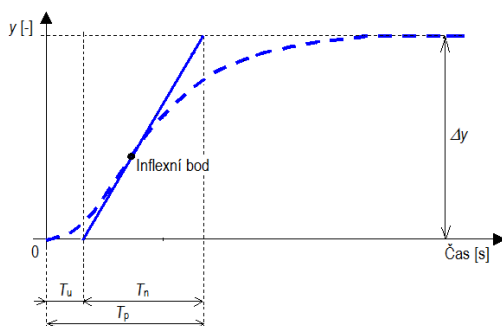
Náhradní přenos má pak tvar

$$G(s) = \frac{k}{T_n s + 1} e^{-T_u s} \quad (17.)$$

ze kterého je pak zřejmé, že přenos má proporcionální tvar se setrvačností prvního řádu a dopravní zpoždění [5]. Koeficient k je dán ustáleným stavem na přechodové charakteristice. Pokud skok akční veličiny není jednotkový skok, ale $\Delta u(t) = \Delta u$ pak

$$k = \frac{h(\infty)}{\Delta u} \quad (18.)$$

Fyzikální rozměr koeficientu přenosu k je dán poměrem fyzikálního rozměru výstupní veličiny $y(t) = h(t)$ k fyzikálnímu rozměru akční veličiny $\Delta u(t)$.



Obrázek 8. Identifikace pomocí T_u a T_n .

Cílem identifikace je získat rovnici popisující chování zvolené soustavy. Přechodová charakteristika je odezva na jednotkový skok $\eta(t)$ (Hevisaidova funkce). Její největší význam je v tom, že ji lze získat velmi snadno měřením.

2.4.2 Simulace

Simulování procesů nám napomáhá jak pochopit tak vyzkoušet, zda dané principy v praxi budou správným řešením, analyzovat, ladit, vyhodnocovat, které řešení je vhodnější a podobně. Simulace je opravdu mocný nástroj a nabývá na důležitosti. Více méně můžeme experimentovat s modelem, který se nám v určité aproximaci chová jako skutečný objekt.

Při vývoji řídicích systémů bývá velmi obtížné a často i nemožné testovat řídicí algoritmy na reálné soustavě, například z důvodu poškození zařízení. Pro simulaci se používá několik typů simulace.

- Model in the Loop
- Software in the Loop
- Processor in the Loop
- Hardware in the Loop

Po ukončení ladění a simulování se provede test na finální soustavě.

2.4.3 Model in the Loop(MIL)

O simulaci MIL hovoříme v případě, je-li matematický model i řídicí systém simulován v simulačním prostředí na jednom PC. Nejedná se zde o real-time simulaci. Model i řídicí algoritmus jsou ve formě „source -code“, není zkompileován. Slouží k předběžnému ověření obrázek 23.

2.4.4 Software in the loop(SIL)

I tato simulace je na jednom PC. V tomto případě, však hovoříme o „target code“, což je zkompileovaná forma kódu. Jedná se zde o real-time simulaci, i když kompilátor i procesor počítače může pracovat odlišně na cílové platformě než na platformě, na které probíhá simulace. Při této simulaci již může být odhalena většina chyb a je možno naladit řídicí systém. Tato situace je na obrázku 21.

2.4.5 Processor in the Loop(PIL)

Při této simulaci se řídicí systém provozuje již na cílové platformě, „embedded target“, v reálném čase. Při simulaci nejsou použity žádné I/O vstupy, čidla či aktuátory, data mezi modelem a modelem jsou posílány přes průmyslovou sběrnici. Tato simulace se používá pro kontrolu výpočetního výkonu hardwaru. V podstatě by mělo být jedno, zda systém reguluje model soustavy či její reálnou podobu.

2.4.6 Hardware in the Loop(HIL)

Simulace HIL je posledním krokem před vývojem prototypu. Stejně jako při simulaci PIL je řídicí systém implementován na konečné platformě. Navíc jsou použity i I/O karty případně emulace čidel a akčních členů. V této simulaci je možné otestovat komplexní řídicí systém včetně DAQ karet a jejich ovladačů.

Odhalení chyby je mnohem levnější ve fázi simulace než na konečném prototypu zařízení. To je hlavním důvodem využití simulace spolu s laděním a analýzou.

2.5 PROFIBUS teorie

2.5.1 Základní informace o síti PROFIBUS.

PROFIBUS (process field bus) je prověřený, široce přijímaný standard pro otevřené sítě typu fieldbus, s podporou celého průmyslového odvětví dodávajícího širokou škálu vybavení, nástrojů a pomoci. Byl vytvořen v roce 1989 německou vládou ve spolupráci s několika výrobci zařízení pro automatizaci. V roce 1993 byl standard přijat jako evropská norma EN 50170 a v roce 2000 byl začleněn do mezinárodního standardu fieldbus, IEC 61158. PROFIBUS je dnes hlavní světový systém sítí fieldbus, s více než 40 milióny nainstalovaných zařízení. V nabídce je zhruba 3 000 výrobků od více než 300 dodavatelů. Celosvětovou podporu poskytuje organizace PROFIBUS International. Regionální asociace PROFIBUS (Regional PROFIBUS Associations, RPA) jsou skupiny uživatelů pracující v rámci PI. Asociace RPA existují v 27 zemích na světě.

Jde o formát zasílání zpráv navržený speciálně pro vysokorychlostní I/O operace ve výrobních závodech a automatizaci budov. Pracuje na bázi otevřeného standardu a je považován za nejrychlejší v současnosti provozovaný prostředek typu FieldBus. Jeho základem je RS485 a evropská elektrotechnická norma EN50170. Přípona DP znamená „decentralizovaná periferie“, což označuje distribuovaná I/O zařízení spojená prostřednictvím rychlé sériové datové linky s hlavní řídicí stanicí. Naproti tomu programovatelný automat (PLC) má standardně vstupní a výstupní kanály uspořádané centrálně. Zavedením síťové sběrnice mezi hlavní řídicí stanicí (masterem) a jeho I/O kanály (slave) se vstupy a výstupy decentralizovaly[10].

PROFIBUS je postaven na mezinárodních standardech a navržen podle referenčního modelu OSI (Open System Interconnection) v souladu s mezinárodní normou ISO 7498. V tomto modelu každá vrstva řeší přesně definované úkoly. Vrstva 1 tohoto modelu je fyzická vrstva a definuje fyzikální charakteristiky přenosu. Vrstva 2 je datová (linková) vrstva, která definuje přístupový protokol sběrnice. Vrstva 7 je aplikační vrstva, která definuje funkce aplikací. Mezi nejrozšířenější varianty komunikačního protokolu PROFIBUS patří:

- PROFIBUS FMS specifikace zpráv v síti fieldbus (Fieldbus Message Specification). Jedná se o původní podobu PROFIBUSU. FMS technologie byla složitá a náročná na implementaci. Proto vzniká jednodušší ale i vylepšená specifikace PROFIBUS DP. FMS již není podporována ale někteří výrobci ji stále používají.
- PROFIBUS DP (Decentralised Periphery) najdeme ve většině zařízení průmyslové automatizace, výrobních linkách a CNC strojích, kde jsou do této sběrnice připojeny řídicí automaty, frekvenční měniče, vzdálené periferie IO, operátorské panely apod. Kabel pro PROFIBUS DP je tvořen dvěma vodiči (červený/zelený), které zajišťují datový přenos o různých rychlostech (např. 500kB, 1,5Mb, 3Mb, 6Mb, 12Mb). Zařízení připojená do sběrnice PROFIBUS DP mají svoje vlastní napájení.
- PROFIBUS PA (Process Automation) se používá v procesní automatizaci a najdeme ho spíše v petrochemických provozech, kde se sbírají data z velkých vzdáleností, kde není kladen důraz na rychlost čtení/zápis hodnoty, kde je výbušné prostředí apod. Na rozdíl od PROFIBUS DP je

protokol PROFIBUS PA přenášen pouze jednou rychlostí 31,25kb a po stejných vodičích je zároveň napájeno připojené zařízení PROFIBUS PA.



Obrázek 9. Znáznornění vrstev podle referenčního modelu OSI.

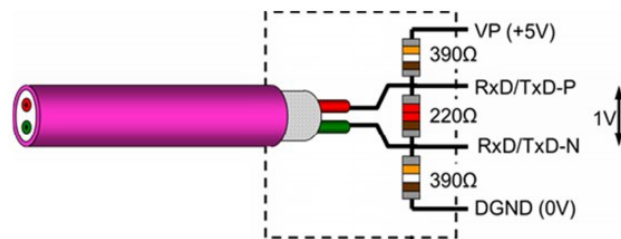
PROFIBUS je velmi silný komunikační protokol, který může fungovat i při chybách v síti, elektromagnetickém zarušení, chybách instalace apod. Jeho vývoj byl zaměřen právě na odolnost proti rušení, chybám a vysokou přenosovou rychlost. Jinými slovy, PROFIBUS Vám přestane fungovat při jedné fatální chybě (např. zkrat) nebo při více malých chybách (špatné zakončení segmentu, zarušení měniči, nadměrně dlouhém kabelu, špatném konektoru apod.) Často pak programátor sníží rychlost sítě nebo zkontroluje všechny konektory a linka se znovu rozjede.[9]

Přenosová technologie:

- metalická síť (vodiče) – RS485 (počet účastníků je omezen na 128 /číslováno 0–127/, telegram délky 11 bitů z toho 8 datových)
- optické vlákno – (s převodníky na rozhraní RS485 lze kombinovat metalický a optický rozvod v jedné síti)
- IEC 1158-2 (proudová smyčka) – (synchronní kódování Manchester)

2.5.2 Topologie sítě

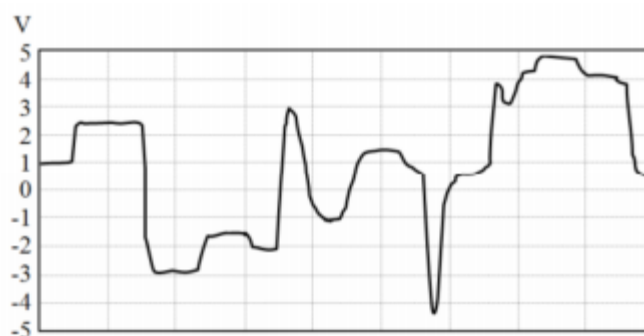
PROFIBUS je založen na komunikačním standardu RS485, který spojuje stanice buď lineárně nebo do hvězdice. Nejčastěji se používá lineární spojení, tzn. stanice jsou propojené PROFIBUS kabelem jedna za druhou a síť je na obou koncích zakončena zapnutými odpory obrázek 10.[9] Zakončení odpory je důležité z důvodu odrazu signálu. Na obrázcích 11,12 jsou průběhy signálu při správném zakončení(obrázek 11.) a při nezakončeném segmentu odpory(obrázek 12).



Obrázek 10. Aktivní ukončení PROFIBUS RS-485



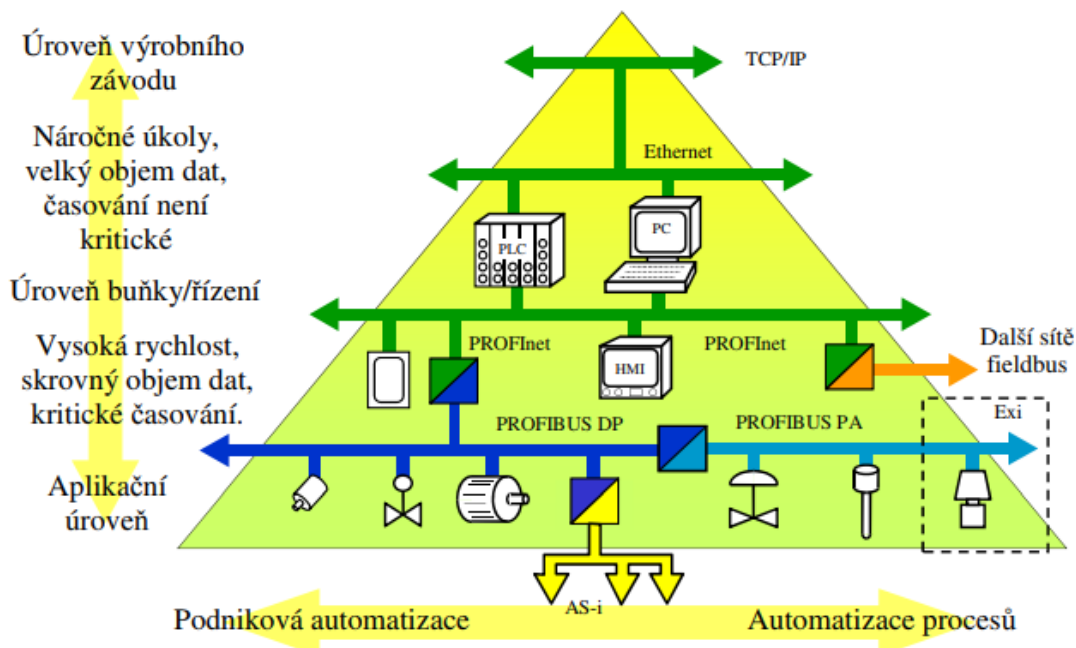
Obrázek 11. Průběh při správném ukončení segmentu



Obrázek 12. Průběh při nesprávném ukončení segmentu

Protokol RS485 byl vyvinut v 60. letech minulého století. Používá se v mnoha různých komunikačních systémech a sítích fieldbus a poskytuje jednoduchou robustní a rychlou komunikaci přes dvoužilový stíněný kabel. Je možné propojit jedním kabelem 32 stanic nebo zařízení. V rámci segmentu se do limitu 32 zařízení započítávají všechny řídicí a podřízené jednotky, opakovače, moduly optického připojení a dokonce analyzátory zpráv. Obrázek 13. nám ukazuje jak PROFIBUS zapadá do hierarchie řídicích systémů. Ethernet se používá na nejvyšší úrovni pro zajištění komunikace v systémech dispečerského řízení a sběru dat, v inženýrských situacích, jako je ladění řídicích systémů a nastavování zařízení. Místo toho PROFIBUS se používá na úrovni nižší, mezi řídicí jednotkou a čidly, motory, ventily atd.

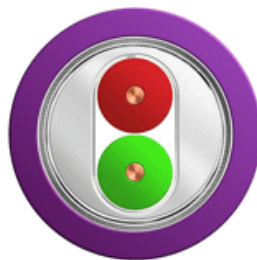
Eternit se v hierarchii pohybuje směrem dolů. Využívá se stále více na úrovni řízení, i aplikace. PROFINET přináší pro tyto účely potřebnou funkcionalitu přes Eternit.



Obrázek 13. Hierarchie řídicího systému

2.5.3 Kabely pro PROFIBUS

Používá se kroucená-stíněná dvojlinka, přičemž stínění slouží zároveň jako uzemnění (průřez obrázek 14.). Dvě hlavní žíly jsou barevně odlišeny zelenou a červenou barvou. Zelený je označován jako linka A (přenos/příjem), červený jako linka B (přenos/příjem). Při připojení konektorů je nutné dodržovat stejné uspořádání kabelů (barev) v celé síti. Různé rychlosti sítě PROFIBUS jsou omezeny maximální délkou jednoho segmentu. Pokud instalace vyžaduje větší délku segmentu, je třeba použít repeater (opakováč). Kabelů PROFIBUS existuje celá řada a liší se zejména způsobem jejich použití a umístění.



Obrázek 14. Průřez standardním kabelem pro PROFIBUS

Univerzální kabel PROFIBUS DP má fialovou barvu a je určen pro běžnou instalaci ve vnitřních prostorech. Ne vždy se kabel pokládá do žlabů, kde je konstantní suché prostředí, stálá teplota, nulové elektromagnetické rušení, žádné mechanické namáhání apod. Z těchto důvodů je třeba zohlednit okolní prostředí kabelu a použít správný typ kabelu pro instalace typu:

- kabel se zvýšenou odolností proti elektromagnetickému rušení (EMC)
- kabel se zvýšenou odolností proti ohybu (pro pohyblivé přívody apod.)
- kabel s ochranou proti hlodavcům
- kabel se zvýšenou odolností proti žáru

- kabel do potravinářských prostor
- a další. [9]

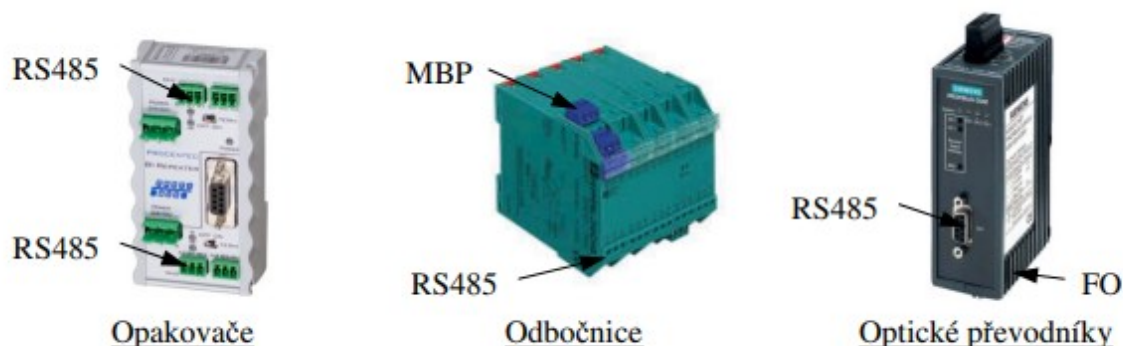
2.5.4 Rozdělení sítě na segmenty

Síť PROFIBUS je tvořena tzv. segmenty, v kterých jsou připojena jednotlivá zařízení. Za jeden segment se považuje takový úsek kabelu PROFIBUS, který je zakončen tzv. zapnutými odpory. Ve většině instalací je použito méně než 32 zařízení a jsou kratší než maximální doporučená délka, proto se v praxi setkáváme více se sítěmi o jednom segmentu.

PLATÍ PRAVIDLO:

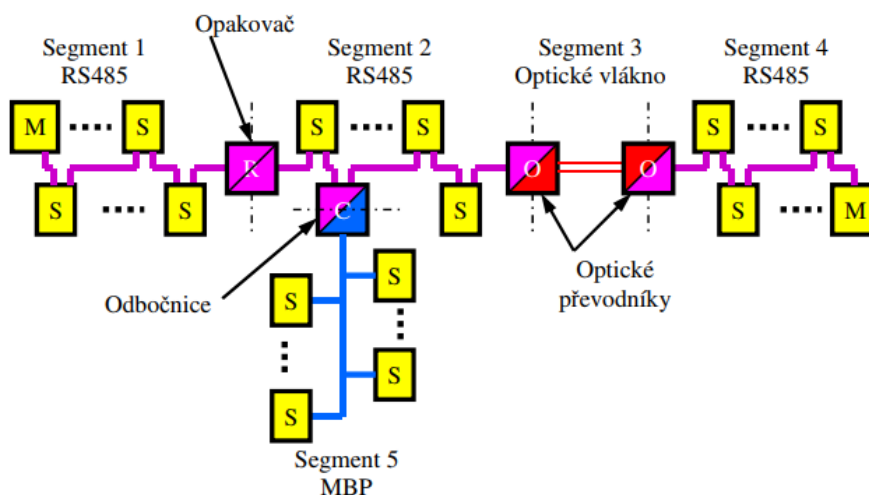
- Maximální počet stanic na jeden segment je 32.
- Maximální počet segmentů je 9 (odděleny např. repeaterem)

Segmenty vznikají pomocí opakovačů, odbočnic nebo optických převodníků.



Obrázek 15. Zařízení zajišťující rozdělení sítě

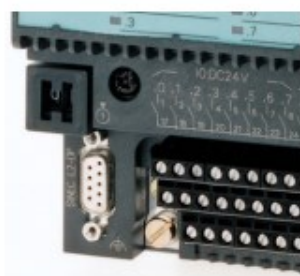
V síti si může jakékoli zařízení vyměňovat informace s jakýmkoli jiným zařízením i když se nachází v jiném segmentu oddělených opakovači, odbočnicemi či opto převodníky viz obrázek 16.



Obrázek 16. Koncept jedné sítě s mnoha segmenty

2.5.5 Konektory

Pro zařízení DP byly definovány standardní 9 pinové sub-D konektory, M12 zástrčky a zásuvky a pro připojení optokabelu hybridní systém připojení.



9pinový sub-D



M12



Hybridní

Obrázek 17. Konektory RS485

Tabulka 1 Přiřazení pinů pro konektory Sub-D

č.pinu	Spojení	Funkce	
1	Stínění	Uzemění	Volitelný
2	M24	Zemění pro 24 V napájení	Volitelný
3	RxD/TxD-P	Datová linka plus (B linka)	Povinný
4	CNTR-P	Ovládací signál směru opakovače	Volitelný
5	DGND	Uzemění dat	Povinný
6	VP	+5 V napájení pro ukončovací rezistory	Povinný
7	P24	+24 V napájení	Volitelný
8	RxD/TxD-N	Datová linka mínus (A linka)	Povinný
9	CNTR-N	Ovládací signál směru opakovače	Volitelný
Pouzdro	Stínění	Uzemění	Volitelný

Tabulka 2 Přiřazení pinů pro konektor M12

č.pinu	Spojení	Funkce	
1	VP	+5 V napájení pro ukončovací rezistory	Povinný
2	RxD/TxD-N	Datová linka mínus (A linka)	Povinný
3	DGND	Uzemění dat	Povinný
4	RxD/TxD-P	Datová linka plus (B linka)	Povinný
5	Stínění	Uzemění	Volitelný
Závit	Stínění	Uzemění	Volitelný

2.5.6 Základní síť PROFIBUS

Zařízení PROFIBUS představuje jakékoli zařízení, které má rozhraní profibus. Jsou to senzory, servomotory, aktuátory, řídicí jednotky, PLC moduly, pohonné jednotky, ventily, čtečky, zařízení pro identifikaci radiových frekvencí, panely operátorských rozhraní, vzdálené vstupně/výstupní jednotky atd., které zpracovává informaci masteru a odesílá svůj výstup zpět do masteru.

Existují dva základní typy zařízení a to hlavní(master) a podřízené(slave). Master řídí celou komunikaci, slave pouze reagují na požadavky hlavních jednotek. Slave tvoří v síti pasivní stanici, protože nemá přístupová práva pro přístup do sběrnice pouze odpovídá na požadavky masteru. Master je aktivní stanice.

Master lze dále rozdělit na:

- Mastr ve třídě 1(Class-1 masters) – neustálá cyklická výměna dat s přiřazenými slaves
- Mastr ve třídě 2(Class-2 masters) – acyklická výměna dat, dle potřeby

Vše ostatní jsou slave zařízení(slave devices).

Data se vyměňují prostřednictvím zpráv neboli telegramů. V jedné síti může být jedna nebo více master jednotek. Master může komunikovat s mnoha slave, ale slave může komunikovat pouze s jednou master jednotkou, je zamknutá ke konkrétní master stanici třídy 1. Master ve třídě 2 může komunikovat s jakoukoli jednotkou ve sběrnici i když je zamknuta k jiné jednotce, jednotku nezamyká jednotku slave. V síti PROFIBUS neexistují priority. Žádný master nemá vyšší prioritu než jiný master, stejně tak to platí mezi slave. Správně nastavená síť zaručuje komunikaci všech zařízení, žádné zařízení se neblokuje. [10]

Master třídy 1 nastaví rychlost přenosu a slave ji automaticky detekují. Master třídy 2 jsou používány na konfiguraci, diagnostiku a podobně, jde například o servisní pc. Mastery se mohou obracet na jednotlivá slave zařízení, jejich definovanou skupinu (multicast) nebo odeslat telegram do všech připojených slave zařízení (broadcast). Slave zařízení vrátí odpověď na všechny telegramy určené jim individuálně, ale neodpovídají na broadcast nebo multicast telegramy z master zařízení. PROFIBUS odesílá zprávy broadcast a multicast jako global control telegramy pomocí adresy 127 a volitelného skupinového čísla cílové skupiny slave zařízení. Protože PROFIBUS používá cyklický způsob dotazování mezi mastery a slave, je také deterministický. To znamená, že chování systému PROFIBUS v čase lze spolehlivě předpovědět. CAN a Ethernet jsou sběrnice systémy řízené událostmi a tvoří proto nedeterministické systémy.

Délka dat přenášených z jednotlivého slave zařízení do master je přesně stanovena v datové bázi slave nebo v souboru GSD. Soubory GSD jednotlivých zařízení připojených prostřednictvím sítě (pouze slave zařízení a mastery třídy 1) dohromady tvoří záznam parametrů mastera, který obsahuje parametrizační a konfigurační data, seznam přidělených adres a parametry sběrnice pro všechny připojené stanice. Master tyto informace používá pro navázání komunikace s jednotlivými slave zařízeními při spouštění.

Ve chvíli, kdy master obdrží záznam parametrů, je připraven vysílat požadavky. Každý slave musí mít platnou adresu pro komunikaci s masterem. Master zašle parametrizační a konfigurační telegramy všem svým přiděleným slave zařízením. Parametrizační a konfigurační telegramy slouží k tomu, aby master znal funkci a konfiguraci slave zařízení. ProfiBus DP nejčastěji používá jedno master zařízení

třídy 1 (mono-master). Přestože se obecně doporučuje jedno masterový provoz, není to povinné. Systém ProfiBus tedy může mít více než jednoho mastera třídy 1, ale komunikace mezi mastery není povolena, s výjimkou případu, kdy jde o udělení přístupových práv ke sběrnici prostřednictvím výměny tokenu.

2.5.7 Adresování

V rámci sítě má každé zařízení svou specifickou přidělenou adresu. Existuje 128 dostupných adres (od 0 po 127). Adresu 127 a 126 nelze přiřadit zařízení. 127 je vyhrazena pro vysílání zpráv a 126 pro pro zařízení, jejichž adresa se nastavuje přes sběrnici. Ostatní jsou volné (0 – 125). Nastavení adres můžeme provést několika způsoby například fyzickým nastavením adresy přepínačem na daném zařízení. Po nastavení adresy je třeba resetovat jednotku pro rozpoznání. Další možností je software nastavení adresy přes PROFIBUS.

2.5.8 Provoz v síti PROFIBUS

Cyklická komunikace master ve třídě 1 a s přidělenými slave, master vysílá požadavek se vstupními daty do první stanice a obdrží odpověď, výstupní data. Dále po odezvě pokračuje do slave 2, slave 3 ... slave n. Poté master začíná znovu slave 1 a proces se opakuje.

Pro komunikaci master se používá metoda předávání tokenů, která kontroluje, aby nehovořilo více jednotek. Je to zpráva, která obsahuje povolení řídit síť. Token vlastní nejprve první master jednotka a ta komunikuje se svými slave. Po komunikaci s posledním slave musí předat token následující jednotce master, aby mohla probíhat komunikace s jejími slave. Komunikovat může pouze jednotka master vlastní token, ostatní musí mlčet a nemohou vysílat žádné požadavky. Po obdržení poslední odpovědi předá master 2 opět token master 1 (v případě dvou masterů), vše se cyklicky opakuje. Fakticky existuje token i v případě jediné jednotky master, která si jej předává sama sobě. Vše výše uvedené probíhá automaticky bez dalšího programování.[11]

2.5.9 PROFIBUS DP slave(Decentralized Periphery)

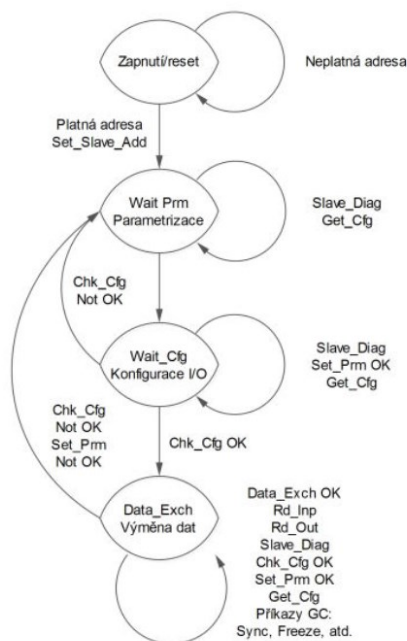
Stavový automat na Obrázek 18 nám popisuje jak PROFIBUS DP funguje ve vztahu ke stavu zařízení. Jsou zde popsány čtyři základní stavy a to zapnutí(reset), parametrizace, konfigurace I/O a výměna dat. Při spouštění používá master tuto sekvenci telegramu:

1. Vyžádání diagnostiky.
2. Změna adresy stanice (volitelně, pouze master třídy 2).
3. Parametrizace slave zařízení.
4. Konfigurace slave zařízení.
5. Opětovné vyžádání diagnostiky před výměnou dat, aby byla jistota, že spuštění systému proběhlo správně.
6. Výměna dat.
7. Global Control (volitelně).

Stav zapnutí/reset:

Stav zapnutí/resetu je počáteční stav po zapnutí napájení slave zařízení DP. V tomto stavu může slave zařízení obdržet telegram od mastera třídy 2, aby změnilo svou adresu stanice. V tomto stavu bude slave zařízení podrženo, pokud nemá platnou adresu od 0 do 125. Po dokončení inicializace při

zapínání, pokud slave zařízení má platnou adresu stanice, přejde zařízení do stavu čekání na parametrizaci.



Obrázek 18. Stavový automat PROFIBUS DP Slave

Stav parametrizace:

V tomto stavu slave zařízení DP očekává parametrizační telegram od mastera, který identifikuje mastera tohoto zařízení a režim, ve kterém má slave zařízení pracovat. Slave zařízení v tomto stavu odmítá všechny ostatní telegramy kromě telegramu s vyžádáním diagnostiky nebo konfigurace. Po nastavení parametrů přejde slave zařízení do stavu konfigurace I/O.

Stav konfigurace I/O:

V tomto stavu slave zařízení očekává konfigurační telegram, specifikující počet bytů vstupů a výstupů, které budou předány v každém cyklu datového telegramu se slave zařízením. Na základě konfiguračního telegramu si slave zařízení také zkontroluje konfiguraci, která mu byla zaslána, podle uložené konfigurace. V tomto stavu slave zařízení přijme telegram s vyžádáním diagnostiky nebo konfigurace nebo telegram s nastavením parametrů.

Stav výměny dat:

Po dokončení parametrizace a konfigurace si slave zařízení cyklicky vyměňuje I/O data s masterem. Jde o cyklický přenos I/O dat a případně diagnostických informací.[11]

V případě výpadku masteru může slave přejít do bezpečného stavu po uplynutí watchdog času bez přijetí správy od masteru. V případě vypršení tohoto času je zřejmé, že master a slave spolu delší dobu nekomunikují a tudíž slave není pod kontrolou masteru. Slave pak opustí režim výměny dat a přejde do předem definovaného stavu. Tento stav je obvykle definován s souboru GSD nebo hardwarově přepínači.

Soubor GSD:

Každý master třídy 1 a každý slave obsahuje soubor GSD. Tento soubor je základním stavebním kamenem záznamu parametrů master. Použitím tohoto souboru konfiguračním nástrojem umožňuje spolupráci plug & play mezi různými zařízeními od různých výrobců. Soubor se nenachází přímo u zařízení, ale obvykle na samostatném disku. Jde o textový soubor v kódu ASCII, který obsahuje data patřící k zařízení. Název GSD souboru je vytvořen z názvu prodejce a identifikačního čísla zařízení. Přípona GSD označuje soubor nezávislý na jazyku. Označení GSE znamená soubor v angličtině a podobně.

Soubor GSD začíná specifikátorem „#Profibus_DP“. V těle souboru jsou parametry specifikovány jako parametry klíčového slova (jako v „keyword = parameter“, viz níže). Komentářové řádky začínají středníkem. Velikost písmen nerozhoduje a na pořadí parametrů nezáleží. Řádky jsou omezeny 80 znaky, ale v psaní na řádek je možno pokračovat tak, že se na jeho konec vloží zpětné lomítko.[11]

Tabulka 3. Rozdělení souboru GSD

Obecné specifikace GSD	Klíčové slovo	Klíčové slovo
Tato část obsahuje informace o názvech prodejce a zařízení, verzích hardwaru a softwaru, ident_číslo, podporované přenosové rychlosti, reakční doby při podporovaných přenosových rychlostech pro sledování časů a volitelně podporu signálů na konektoru sběrnice.	Vendor_name	187.5_supp
	Model_Name	500_supp
	Revision	1.5M_supp
	Ident_Number	MaxTsdr_9.6
	Protocol_Ident	MaxTsdr_19.2
	Station_type	MaxTsdr_93.75
	FMS_Support	MaxTsdr_187.5
	Hardware_Release	MaxTsdr_500
	Software_Release	MaxTsdr_1.5M
	9.6_supp	Redundancy
	19.2_supp	Repeater_Ctrl_Sig
	93.75_supp	24V_Pins
Specifikace slave zařízení v GSD	Klíčové slovo	Klíčové slovo
Tato část obsahuje všechny specifikace týkající se slave zařízení, jako je počet a typ I/O kanálů, specifikace diagnostického textu, podpora automatické detekce přenosové rychlosti, podpora střídavého režimu a u modulárních zařízení informace o dostupných modulech.	Freeze_Mode_supp	Max_Input_Len
	Sync_Mode_supp	Max_Output_Len
	Auto_Baud_supp	Max_Data_Len
	Set_Slave_Add_supp	Unit_Diag_Bit
	User_Prm_Data_Len	Diag_Text
	User_Prm_Data	Unit_Diag_Area
	Min_Slave_Intervall	Module
	Modular_Station	Channel_Diag
	Max_Module	

Specifikace master:

Tuto část má pouze master. Zde jsou maximální počet slave, možnost downloadu a podobně. Soubory GSD všech připojených zařízení tvoří záznam parametrů master. Obsahuje parametrizační a konfigurační data, seznam přidělených adres plus parametry sběrnice. V našem případě obsahuje téměř 4,5 tisíce řádků, na obrázku 19. je začátek našeho GSD souboru.

```

;=====
; GSD-File for ET 200M -1 Siemens AG
; MLFB : 6ES7 153-1AA03-0XB0
; 6ES7 153-1AA02-0XB0
; 6ES7 153-1AA01-0XB0
; 6ES7 153-1AA00-0XB0
; 6ES7 153-1AA83-0XB0
; 6ES7 153-1AA82-0XB0
;
; V 1.26a
;
; Date : 25.08.09 HA, GW
; File : "SIEM801D.GSE"
;=====
; reserved Text Refs : 1 ... 90
; reserved Prm Refs : 1 ... 763
;=====
#Profibus_DP
; =====
; <Prm-Text-Def-List>
; =====
PrmText=2
Text(0)="SIMATIC S7"
Text(1)="SIMATIC S5"
EndPrmText
PrmText=3
Text(0)="No"
Text(1)="Yes (as of ord. no. ...1AA*2-...)"
EndPrmText
PrmText=4
Text(0)="No"
Text(1)="Yes"
EndPrmText
PrmText=5
Text(0)="3 (DC) "
Text(1)="0.1 (DC) "
Text(2)="0.5 (DC) "
Text(3)="15 (DC) "
Text(4)="20 (DC/AC) "
EndPrmText
PrmText=6
Text(0)="2.5"
Text(1)="16.6"
Text(2)="20"
Text(3)="100"
EndPrmText
PrmText=7
Text(0)="Deactivated"
Text(17)="Voltage +/- 80 mV"
Text(18)="Voltage +/- 250 mV"
Text(19)="Voltage +/- 500 mV"
Text(20)="Voltage +/- 1 V"
Text(21)="Voltage +/- 2.5 V"
Text(22)="Voltage +/- 5 V"
Text(23)="Voltage 1..5 V"
Text(25)="Voltage +/- 10 V"
Text(32)="4-wire transducer +/- 3.2 mA"
Text(33)="4-wire transducer +/- 10 mA"
Text(34)="4-wire transducer 0..20 mA"
Text(35)="4-wire transducer 4..20 mA"
Text(36)="4-wire transducer +/- 20 mA"
Text(51)="2-wire transducer 4..20 mA"
Text(66)="Resist. 4-wire conn 150 ohms"
Text(68)="Resist. 4-wire conn 300 ohms"
Text(70)="Resist. 4-wire conn 600 ohms"
Text(130)="Thermal resistor Pt 100 std."
Text(131)="Thermal resistor Ni 100 std."
Text(161)="Thermocouple Intcmp Type N"
Text(162)="Thermocouple Intcmp Type E"
Text(165)="Thermocouple Intcmp Type J"
Text(168)="Thermocouple Intcmp Type K"
Text(177)="Thermocouple Extcmp Type N"
Text(178)="Thermocouple Extcmp Type E"
Text(181)="Thermocouple Extcmp Type J"
Text(184)="Thermocouple Extcmp Type K"
Text(209)="Thermocouple Linintcmp Type N"
Text(210)="Thermocouple Linintcmp Type E"
Text(213)="Thermocouple Linintcmp Type J"
Text(216)="Thermocouple Linintcmp Type K"
Text(225)="Thermocouple Linextcmp Type N"
Text(226)="Thermocouple Linextcmp Type E"
Text(229)="Thermocouple Linextcmp Type J"
Text(232)="Thermocouple Linextcmp Type K"
EndPrmText
PrmText=8
Text(0)="Deactivated"
Text(23)="Voltage 1..5 V"
Text(24)="Voltage 0..10 V"
Text(25)="Voltage +/- 10 V"
Text(34)="Current 0..20 mA"
Text(35)="Current 4..20 mA"
Text(36)="Current +/- 20 mA"
EndPrmText
PrmText=9
Text(0)="Deactivated"
Text(34)="Current 0..20 mA"
Text(35)="Current 4..20 mA"
EndPrmText
PrmText=10
Text(0)="Deactivated"
Text(34)="4-wire transducer 0..20 mA"
Text(35)="4-wire transducer 4..20 mA"
Text(51)="2-wire transducer 4..20 mA"
EndPrmText
PrmText=11
Text(0)="Deactivated"
Text(26)="Voltage +/- 25 mV"
Text(27)="Voltage +/- 50 mV"
Text(17)="Voltage +/- 80 mV"
Text(18)="Voltage +/- 250 mV"
Text(19)="Voltage +/- 500 mV"
Text(20)="Voltage +/- 1 V"
Text(66)="Resist. 4-wire conn 150 ohms"
Text(68)="Resist. 4-wire conn 300 ohms"
Text(70)="Resist. 4-wire conn 600 ohms"
Text(128)="Thermal resist. Pt100 climatic"
Text(130)="Thermal resistor Pt 100 std."
Text(129)="Thermal resist. Ni100 climatic"
Text(139)="Thermal resistor Pt 200 std."
Text(135)="Thermal resist. Pt200 climatic"
Text(131)="Thermal resistor Ni 100 std."

```

Obrázek 19. Ukázka části GSD souboru pro ET200M

Typy přenosu:

- SDA – Posílá data jednosměrně s potvrzením.
- SRD – Posílá data oboustranně v jednom cyklu telegramu. Nejčastěji používáno.
- SDN – Posílá data více stanicím (multicast nebo broadcast) najednou bez potvrzení

PROFIBUS DP využívá pouze SRD a SDN protokolu. U SRD posílá Master výstupní data Slave a vrací se mu vstupní data a to do určitého času. Pokud Slave nemůže odeslat žádné data, odešle „0xE5“. SDN posílá data skupině Slave, nevyžaduje potvrzení.

Formát znaků dat PROFIBUS DP

Všechny znaky obsahují 11 bitů. První bit je START bit, následuje 8 DATA bitů, 1 parity bit (even) a 1 STOP bit. ProfiBus DP provádí výměnu dat v kódu NRZ („Non Return to Zero“, bez návratu k nule).

Pokud se nevysílá na lince je potenciálně nečinného stavu což je logická „1“. Star bit změní tento stav na logickou „0“. Stop bit opět vše vrátí na „1“.

Tabulka 4. Znaky a byty rámce hlavičky telegramu PROFIBUS DP

SD	1 byte	Úvodní oddělovač („Start Delimiter“, slouží k rozlišení formátu telegramu)
LE	1 byte	Čistá délka dat („Net Data Length“) (DU) + DA + SA + FC + DSAP + SSAP.
LEr	1 byte	Délka opakovaně.
DA	1 byte	Cílová adresa („Destination Address“) – kam se zpráva posílá.
SA	1 byte	Zdrojová adresa („Source Address“) - odkud zpráva pochází. Adresa odesílající stanice.
FC	1 byte	Funkční kód („Function Code“) (FC=typ/priorita zprávy). Slouží k identifikaci typu telegramu, tj. požadavek, potvrzení nebo odpověď (FC=13 signály diagnostických dat). Viz dále.
DSAP	1 byte	Cílový přístupový bod služby („Destination Service Access Point“) (COM port přijímače). Používá cílová stanice k určení, jaká služba se má provést.
SSAP	1 byte	Zdrojový přístupový bod služby („Source Service Access Point“) (COM port vysílače).
DU	1 až 32 b (nebo 1 - 244 b)	Datové jednotky („Data Units“) / čistá data od 1 do 244 bytů.
FCS	1 byte	Sekvence kontroly rámce („Frame Checking Sequence“) (doplnění bytů obvodem ASIC v rámci určené délky).
ED	1 byte	Koncový oddělovač („End Delimiter“) (vždy 16H).

Tabulka 5. Znakový rámec v kódu NRZ

Start	D0	D1	D2	D3	D4	D5	D6	D7	Parita	Stop
“0”	0	1	2	3	4	5	6	7	sudá	“1”

Tento znakový rámec platí pro všechny znakové byty, včetně bytů hlavičky telegramu. Při přenosu zpráv v sériových sítích PROFIBUS jsou jednotlivé znakové nebo datové byty posílány v pořadí od nejméně významného bitu („least significant bit“, lsb) k nejvýznamnějšímu bitu („most significant bit“, msb) - viz výše. Při přenosu slova (více než 1 byte) se nejdříve přenáší vyšší byte a potom nižší byte (formát Big-Endian/Motorola).[11] Paritní bit se používá pro kontrolu chyb. Nastaví se na jedna nebo nula podle počtu jedniček v rámci tak aby počet jedniček byl vždy sudý. Je vložen do každého rámce. Přijímací zařízení spočítá počet jedniček a pokud je sudý je vše ok, v opačném případě vyhlásí chybu.

Tabulka 6. Datová rámec hlavičky telegramu(zkratky viz tab. 4)

SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	DU	FCS	ED
1b	1b	1b	1b	1b	1b	1b	1b	1b	var	1b	1b

Úvodní oddělovač(SD) identifikuje začátek telegramu a jeho formát. PROFIBUS DP používá čtyři typy.

Tabulka 7. Úvodní oddělovače

Formát telegramu	Hodnota	Délka datového pole
SD1	10H	0 bytů (není datové pole)
SD2	68H	1 až 32 bytů (nebo až 244)
SD3	A2H	8 bytů pevně.
SD4	DCH	0 bytů (není datové pole)

SD1: telegram s pevnou informační částí a bez datového pole

SD1	DA	SA	FC	FCS	ED
10H	xx	Xx	x	x	16H

SD2: telegram s proměnnou informační částí a délkou datového pole

SD2	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	DU	FCS	ED
68H	X	x	68H	xx	xx	x	3CH	3EH	x..x	X	16H

SD3: telegram s pevnou informační částí a délkou pole

SD3	DA	SA	FC	DU	FCS	ED
A2H	xx	xx	x	x..x	x	16H

SD4: telegram s tokenem master to master

SD4	DA	SA	ED
DCH	xx	xx	16H

Dále existuje krátký potvrzovací telegram

SC	Krátký potvrzovací rámec SC je jednobytová zpráva, která slouží ke kladnému potvrzení požadavku SDA (pouze ProfiBus FMS) nebo zápornému potvrzení požadavku SRD.
E5H	

Master kontaktuje určitý slave tak že umístí 8 bitovou adresu slave do pole DA, do pole SA vloží svou vlastní adresu. Slave vloží do pole SA svou vlastní adresu pro odpověď, aby master věděl, který slave odpovídá. Délka telegramu se uvádí dvakrát(LE, LEr), pro ochranu.

FC specifikuje typ telegramu, typ stanice, prioritu a potvrzení telegramu.

Přístupové body služby (SSAP A DSAP) určuje jaká data se mají přenášet nebo jaká funkce se má provést například SAP = 0(výchozí) cyklická výměna dat, SAP55 změna adresy slave a podobně. Tyto byty používají pouze telegramy SD2, SD3.

Datová jednotka (DU), toto pole obsahuje data pro stanici na DA (data požadavku) nebo data pro stanici na SA (odpovědní data).

Sekvence kontroly rámce (FCS), toto pole obsahuje sekvenci kontroly rámce - součet telegramu (00H..FFH). Jde prostě o součet ASCII bytů informací od DA do DU modulo 256. Kontrolní součet = (DA + SA + FC + DU) mód 256, čili jednoduše sečtení bytů a jejich vydělení FFH (255).

Parametrizace:

Po dokončení inicializace se slave přepne do režimu čekání na parametrizační telegram. Při čekání na parametrizaci slave odmítá všechny telegramy kromě Slave_Diag nebo Get_Cfg.

Telegram Set_Prm – Tato funkce slouží k nastavení parametrů slave zařízení při spuštění, po restartu nebo resetu systému a kdykoliv mimo stav výměny dat. Parametrizační telegram obsahuje minimálně 7 bytů specifických informací vyžadovaných standardem PROFIBUS. Mezi ně patří: monitoring odpovědi/čas watchdogu Twd, čas Tsdr pro časování master/slave, zapnutí/vypnutí režimu Freeze/Sync, zablokování/odblokování slave zařízení pro tohoto mastera, přiřazení skupin, adresa

mastera a identifikační číslo. Telegram může dále obsahovat také další uživatelské parametry v bytech 8 až 32 nebo až do 244.

Parametrizační telegram Set_Prm s hlavičkou:

SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	DU	FCS	ED
68H	X	X	X	8x	8x	X	3DH (61)	3EH (62)	X..	X	16H

Parametrizační telegram Set_Prm DU Byte 1 – Station_status

7	6	5	4	3	2	1	0
Lock_Req	Unlock_Req	Sync_Req	Freeze_Req	WD_On	Vyhrazeno – nast. na 0		
WD_Fact_1							

WD_On je pro aktivaci watchdog funkce která hlídá celkovou délku přenosu. Možností je mnohem více, probrali jsme zde jen ty hlavní.

Konfigurace:

Po úspěšné parametrizaci(Set_Prm) čeká slave na konfigurační telegram(Chk_Cfg). Zde se konfiguruje počet vstupních a výstupních bytů, které je třeba si vyměnit v každém cyklu. Na základě konfiguračního telegramu si slave zkontroluje konfiguraci, která mu byla zaslána. Konfiguraci porovná se zaslaným GSD souborem a v případě shody odešle krátký potvrzovací telegram „E5H“. Master Profibus zjistí konfiguraci vstupních a výstupních dat pomocí telegramu Read_Cfg a potom nakonfiguruje slave zařízení pomocí Chk_Cfg. Odpověď slave zařízení musí obsahovat konfiguraci, s níž se slave zařízení může nabootovat.

Stav výměny dat:

Po dokončení parametrizace a konfigurace může master zahájit cyklickou výměnu I/O dat se slave zařízeními. V režimu výměny dat jsou k dispozici následující služby: Read_Inp (čtení vstupů libovolného slave zařízení), Read_Outp (čtení výstupů libovolného slave zařízení) a Data_Exchange (odesílání a příjem dat slave zařízení, které je parametrizováno a konfigurováno masterem). Slave automaticky zkontroluje přenášená výstupní data, odpoví vstupními daty a v případě zjištění neshody vygeneruje zprávu.

Telegram odesílání výstupních dat nebo přijímání vstupních dat v Data_Exchange

SD	LE	LEr	SD	DA	SA	FC	DU	FCS	ED
68H	X	X	68H	8x	8x	X	X..	X	16H

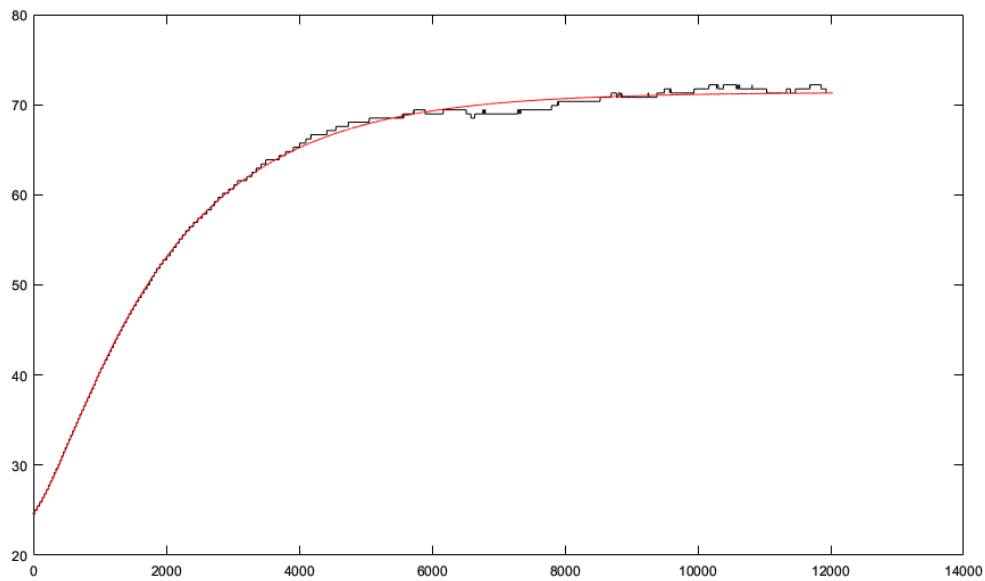
V režimu Data_Exchange master cyklicky odesílá výstupní data na slave zařízení a v odpovědi dostává vstupní data. Pokud je slave čistě výstupní zařízení (nemá žádná vstupní data), odpoví kódem „E5H“. Na rozdíl od všech ostatních telegramů, které mají 11 bytů informací hlavičky, má telegram Data_Exchange pouze 9 bytů, protože používá výchozí SAP (implicitně) a byty DSAP a SSAP nejsou v telegramu obsaženy.

PROFIBUS obsahuje mnohem více dalších možností, funkcí, diagnostických nástrojů. Jedná se tedy o velmi komplexní robusní rozhraní.

3 Praktická část

3.1 Modelování v Matlabu

V první řadě byla změřena přechodová charakteristika tepelné soustavy, což bylo realizováno nastavením vstupní veličiny z 0% na 100%, jednotkový skok, a dále změřen pomocí programu TIA Portal firmy Siemens. Tento software uložil data do souboru excel a odtud byla exportována do výpočetního softwaru Matlab a identifikována pomocí nástroje Identification toolbox. Naměřená přechodová charakteristika je spolu s průběhem identifikace na obrázku 20.



Obrázek 20. Porovnání změřené odezvy(černá) a výsledného modelu(červená) z Matlabu.

Před identifikací je vhodné data upravit filtry pro odstranění rušení, dále posuneme charakteristiku do nulových počátku os odečtením teploty okolí od průběhu. Nyní vložíme data spolu se vstupním signálem do nástroje Identification toolbox. Tento nástroj umožňuje ihned porovnávat vstupní data s vypočtenou přechodovou charakteristikou. Při identifikaci můžeme v tomto nástroji porovnat více možností a nástroj Identification Toolbox nám vypočítá případnou odchylku od experimentálně naměřené přechodové charakteristiky. Výsledný model systému v Laplaceově transformaci:

$$G(s) = \frac{6,57 \cdot 10^{-4}}{s^2 + 0,03115s + 1,403 \cdot 10^{-4}} = \frac{6,57 \cdot 10^{-4}}{(s + 0,02569)(s + 0,00546)} \quad (19.)$$

Jelikož systém bude pracovat na mikrokontroleru Raspberry Pi3 a ten pracuje v diskrétním čase, převedeme rovnici do z transformace se vzorkováním $T=0,1s$. Tento vzorkovací kmitočet je vzhledem k našemu systému dostatečný, jelikož tepelná soustava má poměrně pomalou odezvu. Nejedná se o rychlý děj.

Dostaneme tedy:

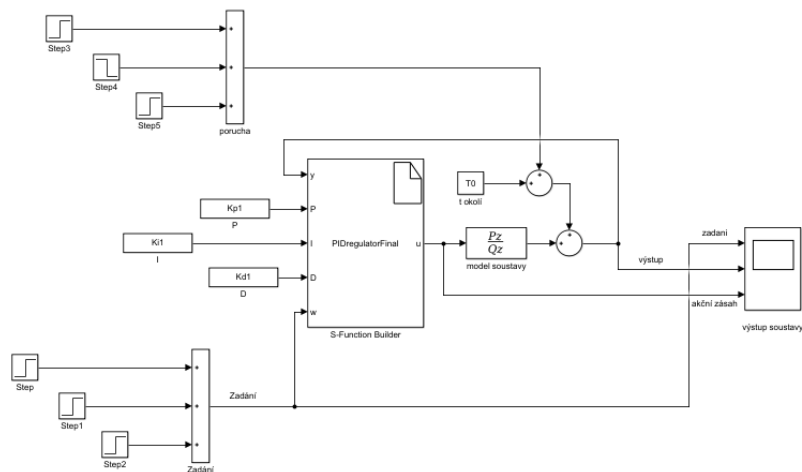
$$G(z) = \frac{3,282z \cdot 10^{-6} + 3,27 \cdot 10^{-6}}{z^2 - 1,997z + 0,9969} \quad (20.)$$

Tuto přechodovou charakteristiku můžeme vložit do našeho modelu soustavy obrázek 21.

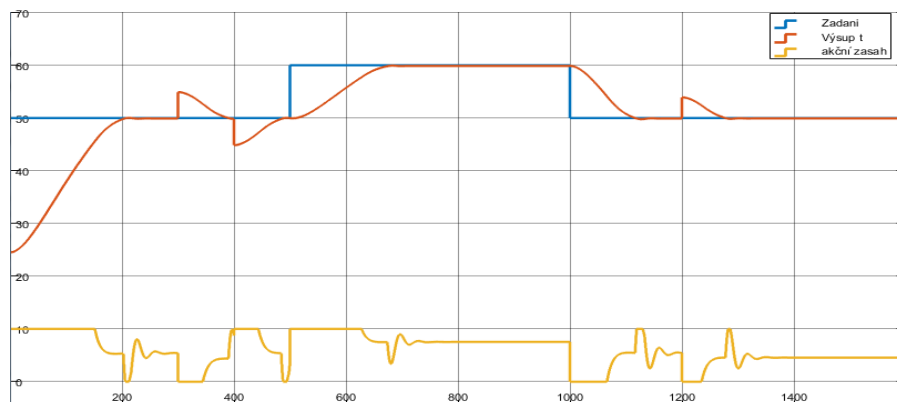
3.2 Simulace

Aby regulátor plně odpovídal požadavkům aplikace, je třeba implementovat algoritmus PID v diskrétní podobě do mikrokontroleru. K tomu je zapotřebí rovnice, kterou pak v samotném regulátoru použijeme. Zde byl zvolen paralelní model obrázek 27. Pro samotnou simulaci kódu byla využita S-funkce v prostředí simulink. Nejprve byly nadefinovány vstupy a výstupy a vložen vytvořený kód PSD regulátoru v podobě kódu v programovacím jazyce „C“, určený pro náš regulátor. S-funkce umožňuje zkompilovat kód regulátoru a vložit jej v prostředí simulink do bloku. Aby byla možnost PSD konstanty měnit, jsou zde vyvedeny jako vstupy, podobně zadaná hodnota w . Výstupem je samotný akční zásah, vcházející do modelu obrázek 21. Použita byla i možnost zavedení poruchové veličiny pro kontrolu správné činnosti regulátoru (změna teploty okolí, rušení, termoelektrický jev na svorkovnicích a podobně).

Jedná se o simulaci SIL, kdy je matematický model a řídicí systém modelován na jednom počítači v našem případě v prostředí Matlab, algoritmus regulátoru je zkompilován v podobě „target code“. Bloky Step-step2 zavádíme do modelu požadované zadání teploty soustavy, step3-step6 zavádí do modelu poruchu.



Obrázek 21. Schéma v simulinku



Obrázek 22. Výstup soustavy, akční zásah a zadání

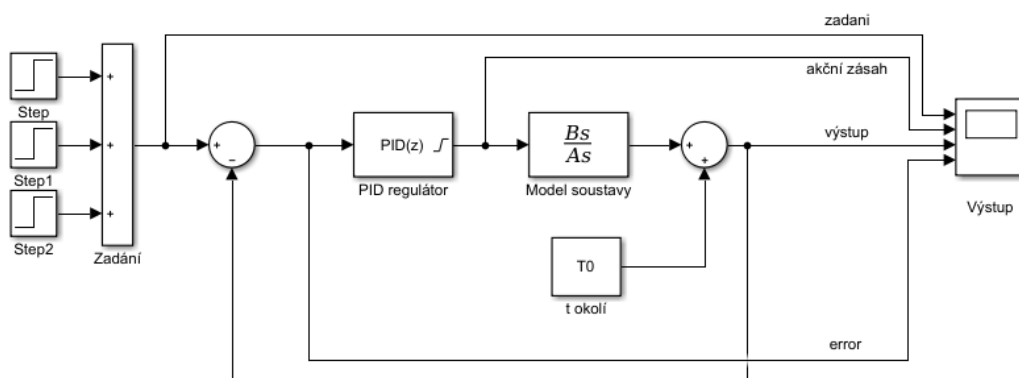
Výsledná teplota (červená) nám správným způsobem kopíruje zadání (modrá) viz obrázek 22.

Odchylky v čase 300, 400 a 1200 jsou způsobeny úmyslným zavedením poruchy. Je patrné následná reakce akčního zásahu a dotažení teploty, což bylo naším úkolem. Akční zásah (žlutá) nám nekmitá a poměrně rychle se nám ustálí.

3.2.1 Ladění soustavy

Dříve než spustíme samotnou simulaci je třeba celý model naladit. K tomuto účelu slouží PID konstanty. Jak už bylo dříve řečeno, více než 80% regulačních obvodů je nesprávně naladěno. Správné naladění je velmi důležité, protože určuje zda ve výsledku bude mít regulátor požadované parametry a při špatném nastavení nemusí pracovat vůbec a v případě nejhorším může dojít až ke zničení soustavy (například při rozkmitání ventilu) čehož se v našem případě nemusíme obávat. Tepelná soustava je poměrně stabilní a ani rozkmitání akčního zásahu by nemusel znamenat vážný problém.

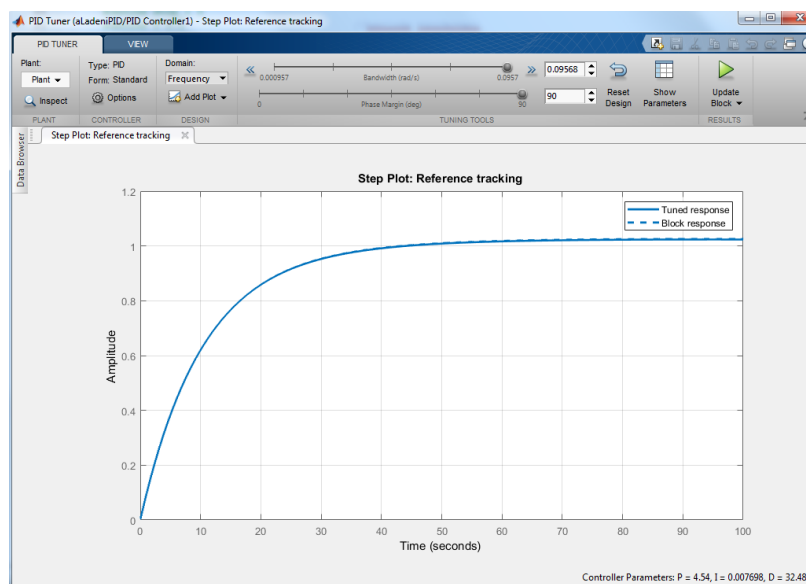
K tomuto účelu obsahuje Matlab nástroj tune, který nám vypočítá požadované PID konstanty.



Obrázek 23. Blokové schéma pro ladící model v simulinku

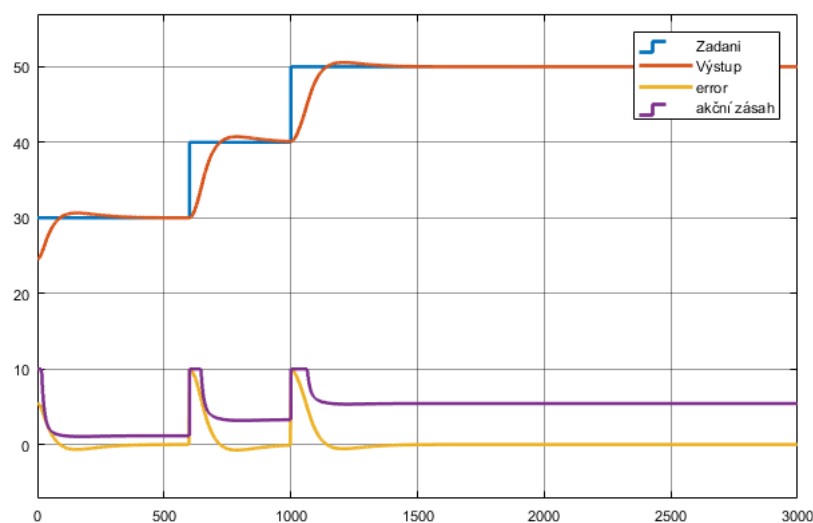
Je ovšem nutné, zvolit správný model v bloku PID. Při nesprávně zvoleném modelu nám nástroj Tune bude vypočítávat odlišné konstanty a model nebude pracovat správně. Zvolil jsem regulátor s vytknutou proporcionální složkou dle rovnice $P(1 + IT_s \frac{1}{z-1} + D \frac{1}{T_s} \frac{z-1}{z})$, s nastavením Limit output na 0 – 10 a použitím Anti-windup metod na volbu clamping. Nastavení limitu odpovídá našemu výstupu 0-10 V. Anti-windup je dobrým způsobem zrychlení ustálení a stability, neboť zabraňuje přebuzení integrační složky. Ta nám čistě matematicky může dávat hodnoty mnohem vyšší, než je

reálný akční zásah a snižovat se začne až po překmitnutí, což může dlouho trvat. Proto ji omezíme na maximum výstupu, čímž se zmenší překmit systému. Pro správné naladění PID konstant je nutno také vložit model soustavy stejně jako u simulace. Nyní můžeme v bloku PID regulátor kliknout na nástroj Tune a začít samotné ladění soustavy. Zobrazí se charakteristika v časové oblasti, a posuvníky Response Time a Transient Behavior máme možnost upravit průběh dle našich představ. Tlačítkem Update Block konstanty uložíme obrázek 24. Po naladění konstant můžeme zkontrolovat průběhy výstupu a akčního zásahu. Jsme-li spokojeni z výsledkem průběhů, můžeme konstanty použít pro náš simulační model viz výše. Průběhy jsou naprosto stejné jako v případě ladění, regulátor je nalaďen a pracuje dle našich požadavků.



Obrázek 24. Ladicí nástroj Tune Toolbox

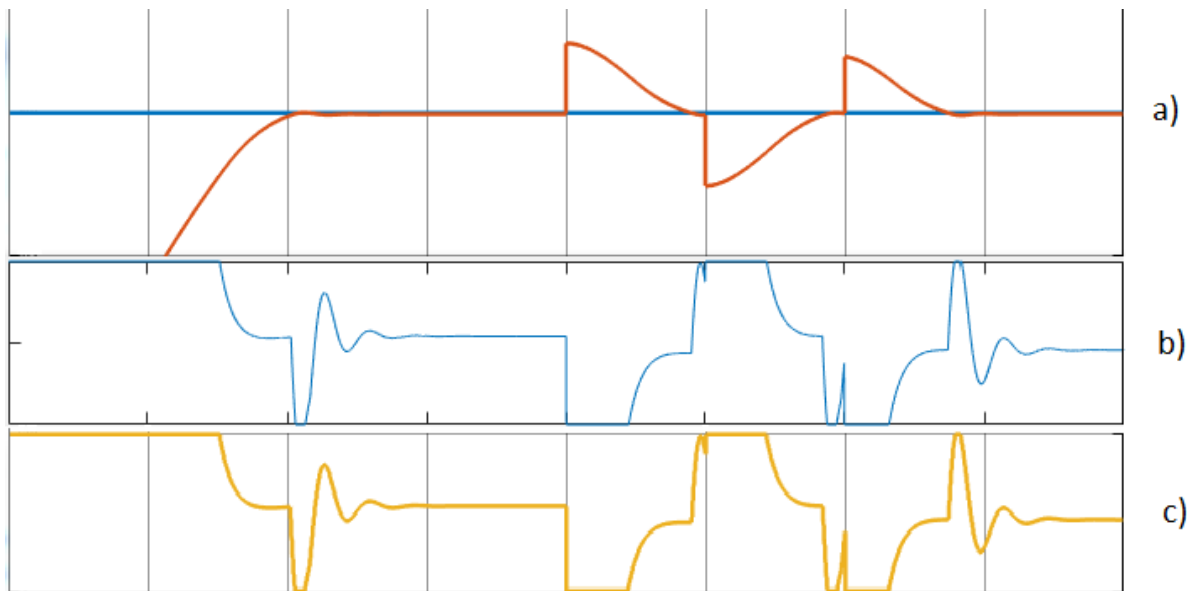
Zde se jedná o simulaci MIL. Kód regulátoru není zkompileován, je simulován blokem PID v simulinku, což můžeme snadno využít k ladění a nastavení PID konstant. Výsledkem je graf na obrázku 25. kde vidíme jak výstupní hodnota sleduje zadání.



Obrázek 25. Naladěná soustava PSD regulátoru.

3.2.2 Simulace v Raspberry Pi3

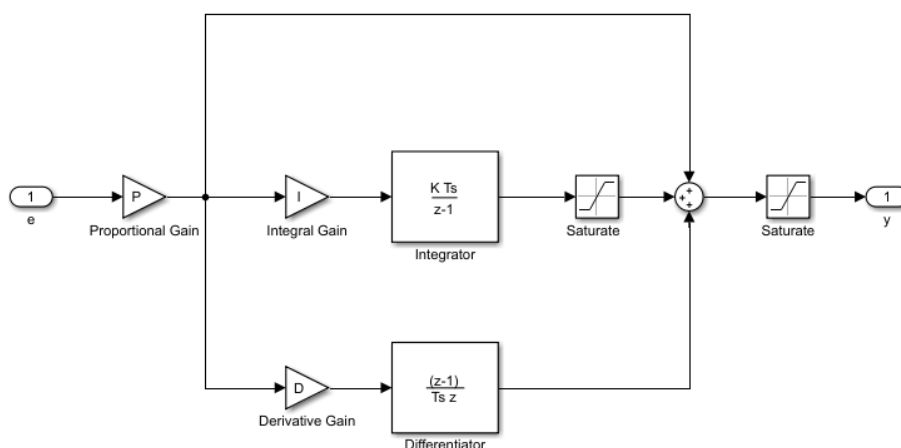
Po úspěšné simulaci v simulinku byl algoritmus regulátoru nahrán na cílovou platformu Raspberry Pi3. Jako vstupní data byly využity vstupní a výstupní data ze simulace SIL v prostředí Matlab. Zadání jsou vstupní data, výstup je zaváděn do vstupu pro výpočet regulační odchylky. Jedná se o simulaci PIL s otevřenou smyčkou. Tímto způsobem ověříme, že regulátor funguje stejně jako v případě naší simulace SIL.



Obrázek 26. Porovnání průběhů simulací, a) zadaná hodnota(modře)+výstupní teplota(červeně) vstupní data pro simulaci, b) výstup ze simulace PIL, c) výstup ze simulace SIL

Jak je vidět z obrázku 26. data b) a data c) jsou totožná. Regulátor nám funguje dle našich představ a na platformě Raspberry Pi funguje naprosto stejně jako v prostředí Matlab.

3.3 Algoritmus PID regulátoru



Obrázek 27. Schéma PID regulátoru

Při návrhu algoritmu PSD regulátoru jsem vycházel s blokového schéma na obrázku 25. Jelikož se pohybujeme v diskretním čase je integrace nahrazena sumou a derivace nahrazena diferencí, výsledný algoritmus má pak tvar:

Algoritmus regulátoru v jazyce C:

```
{
error = w - y;

proporce = P * error;

suma += (T/ I)* proporce;

deriv = D/T * (proporce-proporceLast)

if (suma>umax) suma=umax;           //omezení integrační složky
if (suma<umin) suma=umin;

u= proporce + suma + deriv;

if (vystup>umax) vystup=umax;       //saturace výstupu
if (vystup<umin) vystup=umin;

proporceLast = proporce;

return u;
}
```

P, I, D – konstanty regulátoru

T – perioda vzorkování

w – zadaná hodnota

y – změřená hodnota

Tento algoritmus je pak vložen do S-funkce viz výše simulace.

umin – konstanta nastavena na hodnotu 0

umax - konstanta nastavena na hodnotu 10

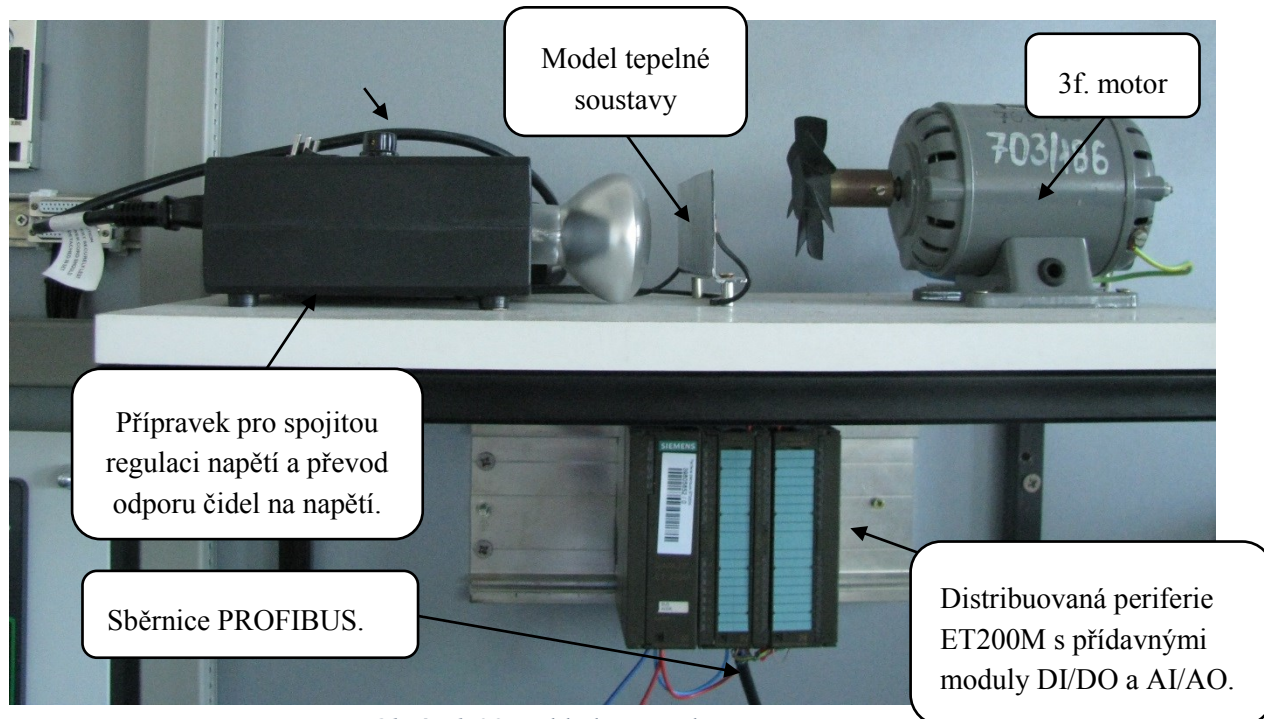
3.4 Implementace

Pro nainstalování operačního systému do Raspberry Pi3 byl použit balíček NOOBS, který kromě základního systému Raspbian nabízí i další operační systémy. Systém NOOBS byl stažen z oficiálních stránek <https://www.raspberrypi.org/downloads/noobs/> a po rozbalení pomocí diskového manageru Win32DiskImager nahrán na SD kartu určenou pro systém Raspberry Pi3. SD karta byla vložena do Raspberry a spuštěn systém.

Po odladění PSD regulátoru je nutné, buď použít croscompiler a zkompileovaný soubor přenést na Raspberri, nebo přenést kód na Raspberry a zkompileovat jej zde na cílové stanici. Takže kód byl přenesen na Raspberry a zkompileován pomocí příkazu „gcc -pthread -o regulator regulator.c“. GCC (GNU C Compiler) je použitý kompilér který je přímo obsažen v systému NOOBS.

3.4.1 Realizace soustavy

Naše soustava obsahuje Distribuovanou periferii ET200M s přídatnými moduly DI/DO a AI/AO, dále Frekvenční měnič Micromaster 420. Přídatné moduly analogových a digitálních I/O jsou připojeny k tepelné soustavě se žárovkou a motorkem.

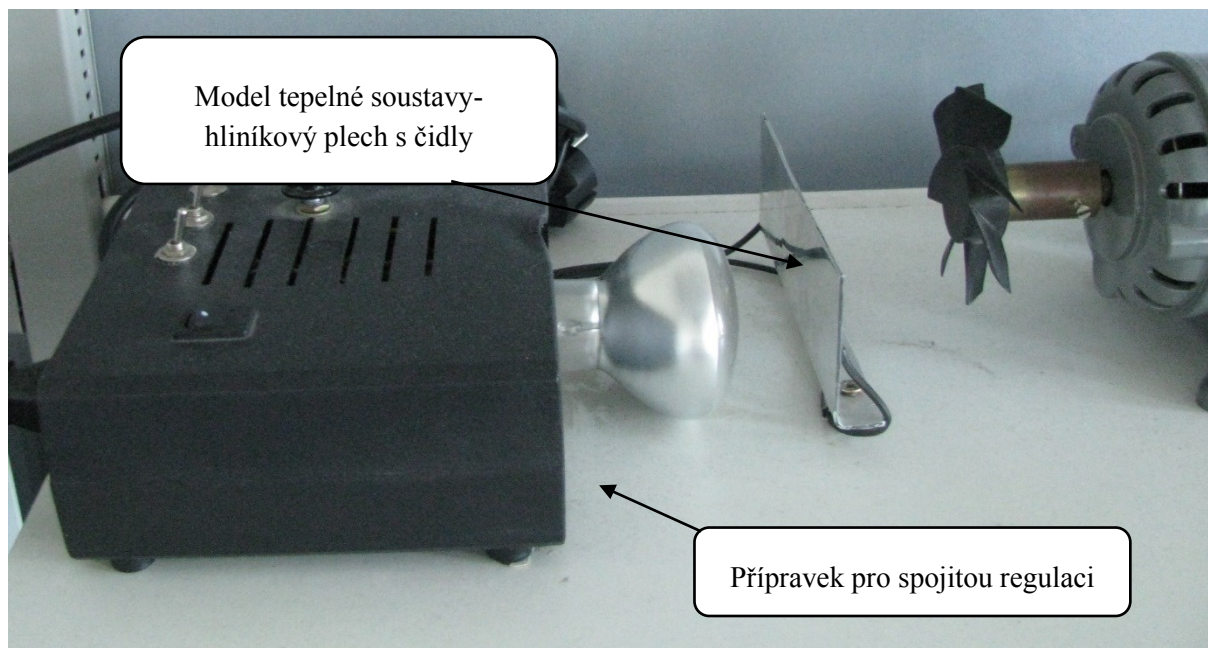


Obrázek 28. Pohled na tepelnou soustavu.

Jak ukazuje obrázek 40. Raspberry Pi je použit jako master, následuje převodník USB/RS485, kabeláž, Slaves v podobě vzdálené periferie siemens ET200M a Mikromáster 420 pro pohon třífázového motoru, akční členy je přípravek z laboratoře jenž řídí halogenovou žárovku, tepelná soustava je realizována hliníkovou destičkou, a čidlo snímající teplotu. V druhém konektoru je vedení zakončeno odporovým můstkem dle obrázku 10., z důvodu odrazu signálu. Spojení jednotlivých částí je realizováno devíti pinovými konektory SUB-D a dvoužilovými stíněnými kabely.[21]

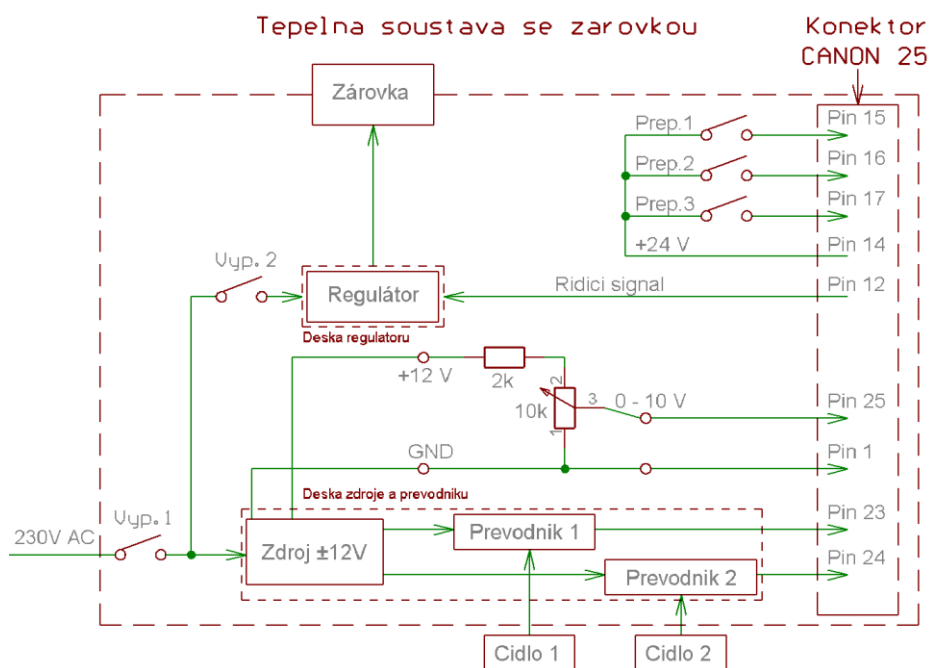
3.4.2 Tepelná soustava se žárovkou a motorkem

Pro spojitou regulaci osvětlení žárovkou je použit přípravek spojité regulace napětí a převod odporu čidel na napětí. Regulaci osvětlení je řízena externím signálem o úrovni 0-10V. Na hliníkovém plechu viz obrázek 29. jsou umístěna dvě odporová čidla snímající teplotu modelu tepelné soustavy.

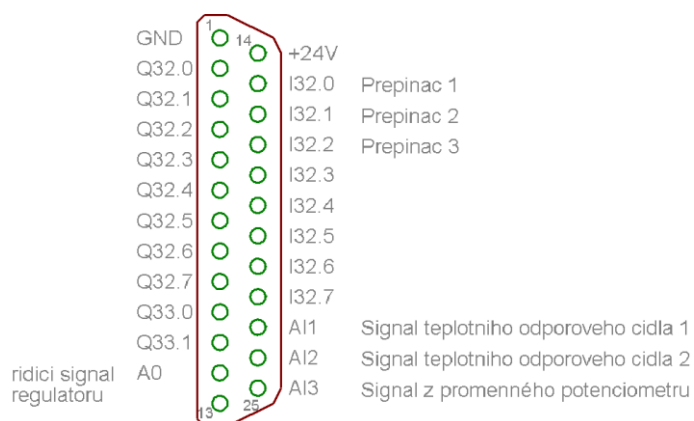


Obrázek 29. Tepelná soustava spolu s žárovkou, motorkem a přípravkem pro spojitou regulaci

Na obrázku 30. je blokové schéma našeho přípravku pro spojitou regulaci. Soustava je napájena síťovým napětím 230V, které je spínáno vypínačem Vyp.1, který plní funkci hlavního vypínače. Za vypínačem je síťové napětí přivedeno na napájecí zdroj pro převodníky a také přes další vypínač Vyp.2 pro spojitý regulátor výkonu žárovky 100 W. Signály z jednotlivých čidel jsou v převodnících zpracovány a vyvedeny na konektor CANON 25 pin 23,24. Spojitý regulátor výkonu žárovky je možné samostatně vypnout pomocí vypínače Vyp.2 a řídí se signálem přivedeným z konektoru CANON 25 pin 12.[21]



Obrázek 30. Blokové schéma přípravku pro spojitou regulaci



Obrázek 31. Zapojení vývodů konektoru CANON 25.

Tabulka 8. Seznam technologických I/O signálů

Název	I/O	č. pinu	Symbol	Rozsah	Typ
Analogové vstupy					
Teplota – 1	AI 0	23	θ_1	$(0 \div 100)^\circ\text{C} \approx (0 \div 10) \text{ V}$	Teploměr KTY83-110
Teplota – 2	AI 1	24	θ_2	$(0 \div 100)^\circ\text{C} \approx (0 \div 10) \text{ V}$	Teploměr KTY83-110
Potenciometr	AI 2	25	Pot.	$0 \div 10 \text{ V}$	Hodnota potenciometru
GND		1			
Analogový výstup					
výkon topení	AO 0	12	výkon	$(0 \div 100)\% \approx (0 \div 10) \text{ V}$	regul. topení K8003
GND		1			
Digitální vstupy					
tlačítko 1	I 32.0	15	tl. 1	24VDC	Přepínač 1
tlačítko 2	I 32.1	16	tl. 2	24VDC	Přepínač 2
tlačítko 3	I 32.2	17	tl. 3	24VDC	Přepínač 3
+24V		14			

3.4.3 Obvod spojitě regulace

Jedná se o modul firmy Vellman označen F-KV8003. Regulátor je ovládán stejnosměrným napětím 0-10V. Regulační vstup je galvanicky oddělen optočlenem a proto je vhodný pro dálkové ovládání síťového napětí z výstupu různých převodníků napětí nebo výstupu z PC.[21]

Technické údaje:

Regulované napětí 110 nebo 220/240 V AC, 50 nebo 60 Hz.

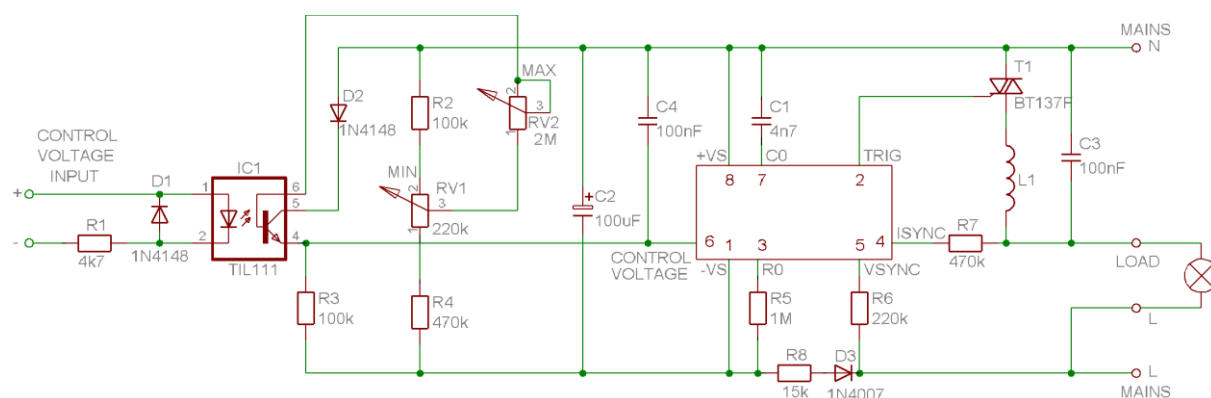
Max. regulovaný proud 3,5 A / 750 W / 220 V.

Napětí pro řízení 0 – 10 V DC.

Max. řídicí proud 2,25 mA.

Opticky izolovaný vstup řídicího napětí.

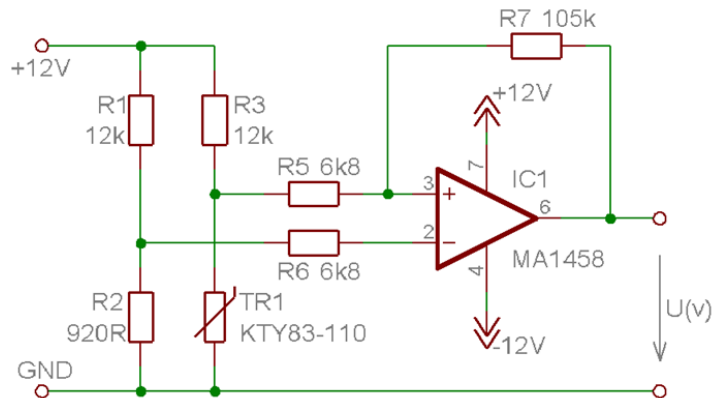
Triak je na chladiči izolován.



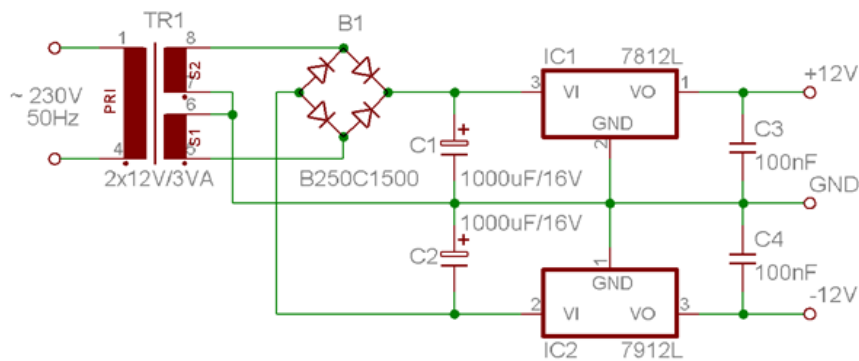
Obrázek 32. Schéma stovebnice Vellman s označením F-KV8003.

3.4.4 Převodník k odporovým teplotním čidlům

Odporové teploměry použité v sestavě jsou připojeny přes jednoduchý převodník. Čidlo tvoří spolu s odpory odporový dělič připojený na napájecí napětí, výstup je přiveden do operačního zesilovače MA 1458. Kdyby teplotní čidlo mělo lineárnější charakteristiku, bylo by vhodné jej napájet ze zdroje konstantního proudu. Avšak charakteristika udávaná v katalogu má nelineární průběh, proto je snímač zapojen v odporovém děliči, který mírně jeho charakteristiku napřimuje. Druhý odporový dělič představuje zdroj referenčního napětí, které kompenzuje vzniklý napětový offset na 0 V při teplotě 0 °C. Převodník je napájen stejnosměrným stabilizovaným zdrojem napětí s pevně nastaveným výstupním napětím ± 12 V realizovaný pomocí obvodů 7812 a 7912 zapojených podle katalogových doporučení. Jedná se o kladný(7812) a záporný(7912) stabilizátor.[21]



Obrázek 33. Schéma převodníku teplotních čidel



Obrázek 34. Schéma Stabilizovaného zdroje napětí

3.4.5 Micromaster 420

MICROMASTER 420 je měnič kmitočtu s napětovým meziobvodem určené pro napájení třífázových asynchronních a synchronních elektromotorů ve výkonovém rozsahu od 120 W do 11 kW. Podle typu měniče je možné jednofázové nebo třífázové napájení.

Obvody řízení a regulace jsou realizovány pomocí digitální techniky s mikroprocesorovým řízením a výkonovými tranzistory typu IGBT. To činí měniče spolehlivými zařízeními s možností přizpůsobení vlastností velkému množství aplikací. Metodou pulzně šířkové modulace a přepínatelným spínacím kmitočtem je dosaženo tichého a rovnoměrného chodu motoru. Ochranné funkce měniče 3-fázového motoru zajišťují dokonalou ochranu pohonu.



Obrázek 35 3.4.5 Micromaster 420

Tovární nastavení měniče je vhodné pro široký rozsah jednoduchých aplikací. Měníče je možné použít jako samostatná zařízení nebo je začlenit do automatizačního celku.[21]

Základní vlastnosti

- Velice snadné připojení, nastavení a uvedení do provozu.
- Rychlá odezva na řídicí signály.
- Obvody mikroprocesorového řízení a regulace zabezpečují vysokou spolehlivost a flexibilitu zařízení.
- Množství parametrů umožňuje dokonalé přizpůsobení pohonu s měničem kmitočtu dané aplikace.
- Vysoký spínací kmitočet pulzně šířkové modulace zajišťuje tichý chod motoru.
- Možnost výběru způsobu ovládání přes řídicí svorkovnici se základním ovládacím panelem (BOP), rozšířeným ovládacím panelem (AOP), sériovou linkou z PC nebo komunikační sběrnici PROFIBUS.
- Požadovanou hodnotu výstupu kmitočtu (a tedy i hodnotu otáček motoru) lze ovládat těmito způsoby:
 1. přímým číselným zadáním hodnoty kmitočtu,
 2. analogovým napětovým signálem,
 3. externím potenciometrem,
 4. motorpotenciometrem,
 5. až sedmi pevně přednastavitelnými hodnotami kmitočtu,
 6. prostřednictvím sériového rozhraní (USS protokol, PROFIBUS).
- Přednastavené hodnoty parametrů odpovídají požadavkům evropských a severoamerických norem.

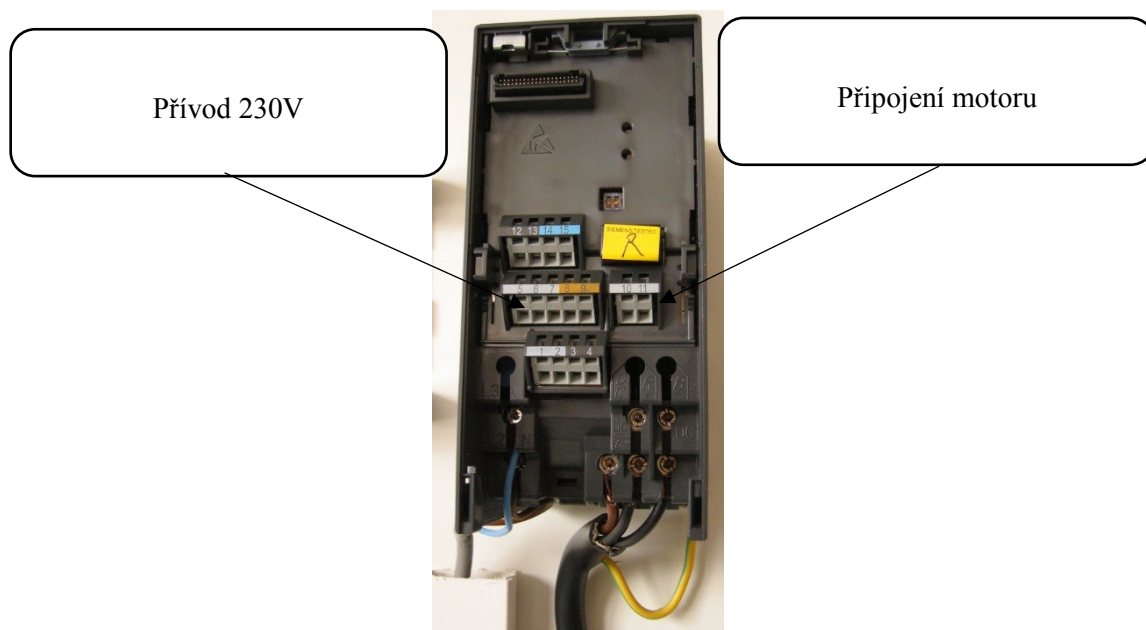
Rozšířené vlastnosti

- Řízení s aktivní regulací magnetizačního proudu (FCC) pro dynamicky náročné aplikace.
- Rychlá ochrana proti nadměrnému vzrůstu proudu (FCL) umožňuje reakci měniče dříve tak, aby nedošlo k vyhodnocení poruchového stavu a zastavení pohonu.
- Možnost brždění motoru stejnosměrným proudem.
- Kompaundní způsob brožení motoru umožňuje řízené zastavení pohonu s velkým momentem setrvačnosti.
- Nastavitelná doba rozběhu i doběhu s počátečním i koncovým zaoblením rozběhové křivky pro měkký rozběh a zastavení pohonu.
- Vestavěný technologický PI regulátor umožňuje řízení procesu bez nutnosti nadřazeného řídicího systému. [21]

Ochranné funkce

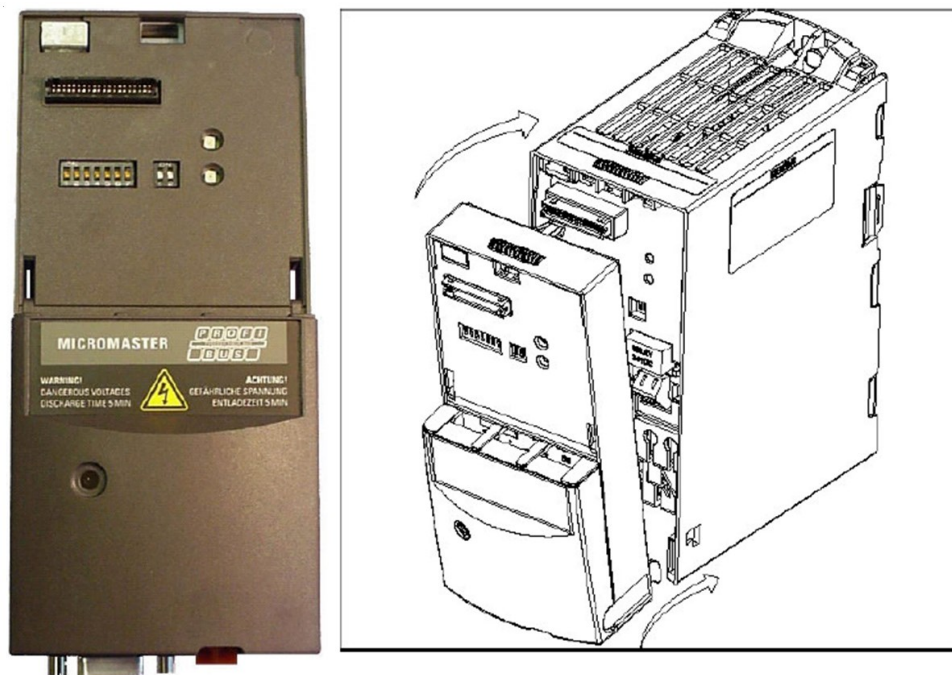
- Kompletní ochrana měniče i motoru před přetížením.
- Ochrana proti přepětí a podpětí.
- Ochrana proti překročení teploty měniče i motoru.
- Tepelná ochrana I^2t motoru.
- Ochrana proti zemnímu a mezifázovému spojení. [21]

Na obrázku 35. je po odejmutí čelního panelu vidět silové připojení sítě a motoru.



Obrázek 36. Micromastr 420 po odpojení čelního panelu

Nastavení mikromastu pro komunikaci PROFIBUS sběrnice je na Obrázku 36.



Obrázek 37. Připojení rozšiřujícího modulu pro sběrnici PROFIBUS

Na rozšiřujícím modulu k PROFIBUS je umístěna signalizační LED dioda, u které můžeme podle indikace zjistit v jakém stavu se měnič nachází. Jestli přijímá data nebo jestli je stavu poruchy a jiné. Před provozem musíme nastavit spoustu parametrů určujících vlastnosti motoru i vlastní způsob komunikace přes sběrnici PROFIBUS dle následujícího postupu. [21]

Tabulka 9. Nastavení Mikromasteru 420

Číslo parametru	Název parametru	Rozsah hodnot	Tovární nastavení	Nastavení uživatelem	Komentář
P0003	Přístupová práva	0 - 4	1	3	určuje administrátorský přístup
P0918	Adresa měniče na sběrnici PROFIBUS	1-125	3	15	nastavujeme pomocí přepínačů DIP
P0700	Způsob ovládání měniče	0 - 6	0	6	nastavuje sériovou linku PROFIBUS / Fieldbus (s komunikačním modulem)
P1000	Výběr zdroje žádané hodnoty	0 - 66	2	6	hlavní žádaná hodnota přes komunikační linku PROFIBUSu bez přídavné hodnoty
r2050	Přijátá data PZD komunikační linky PROFIBUS	-	-	-	Zobrazení přijatých dat části PZD telegramu komunik. Linky PROFIBUS
P2051	Vysílaná data PZD komunikační linky PROFIBUS	-	-	-	Zdroj vysílaných dat části PZD telegramu sériové linky USS1
P2041	Parametry PROFIBUS	0 - 65535	0	-	Speciální data vysílaná po komunikační lince PROFIBUS
P2040	Maximální přístupová prodleva mezi dvěma po sobě jdoucími telegramy PROFIBUS	0 - 65535ms	20ms	-	0 deaktivuje funkci hlídání prodlevy mezi dvěma telegramy
P0927	Povolení zařízení pro změnu parametru	0 - 15	15	15	podle parametru je povoleno/zakázáno měnit hodnoty parametrů z určitého zařízení - nastavuje se komplikovaněji - třeba se podívat do manuálu k MM4 do článku 4.2 Popisu parametrů str.69
r0254	Diagnóza komunikačního modulu	-	-	-	viz. Manuál k rozšiřujícímu modulu článek 7.3 Diagnóza pomocí parametrů diagnózy str.65

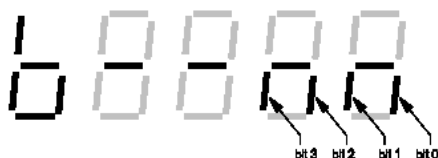
Nejprve je třeba nastavit P0003 na 3, které nám dovolí přístup na všechny ostatní parametry, které běžně nejsou přípustné.

Adresa měniče pomocí P0918 se nemusí zadávat, pokud je již nastavena pomocí DIP přepínačů na komunikačním rozšiřujícím modulu PROFIBUS. Jsou-li přepínače přepnuty na 0 potom je adresa brána z parametru P0918.

číslo bitu	segment svítí	segment nesvítí
bit 0	komunikační modul PROFIBUS POVOLEN	ZAKÁZÁN
bit 1	ovládací panel BOP, AOP POVOLEN	ZAKÁZÁN
bit 2	sériová linka USS1 (RS232) POVOLENA	ZAKÁZÁNA
bit 3	sériová linka USS2 (RS485) POVOLENA	ZAKÁZÁNA

Výsledná hodnota parametru je dána součtem následujících hodnot

- 1 (bit 0)
- 2 (bit 1)
- 4 (bit 2)
- 8 (bit 3)



Obrázek 38

P0700 a P1000 musí být nastaveno na hodnotu 6, abychom mohli přes komunikační rozšiřující modul přijímat a odesílat data. Parametr P0927 musí být dle dokumentace nastaven nejlépe na hodnotu 15. Tabulky jsou nutností ke správnému zprovoznění komunikace PLC s MICROMASTERem. Veškeré toto konfigurování se provádí přes tlačítka na ovládacím panelu BOP viz obrázek 38. [21]

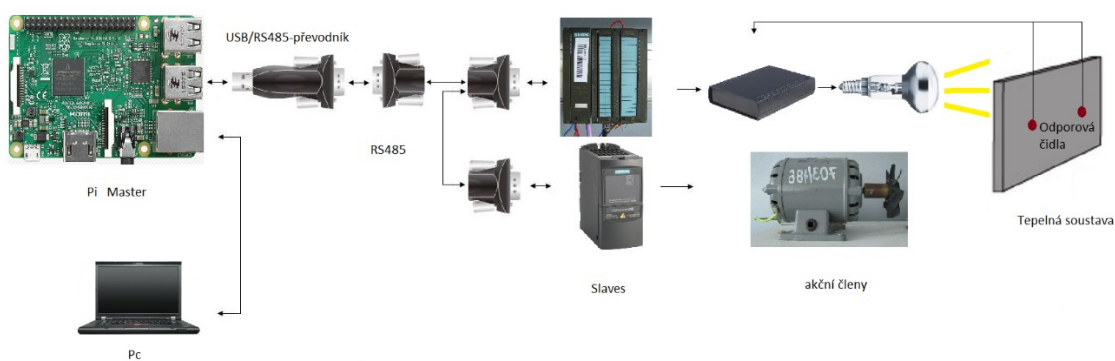


Obrázek 39 Ovládací panel BOP

Ovládací panel BOP je doplňkem měniče a umožňuje uživateli přístup k parametrům měniče a jejich změnou přizpůsobením měniče různým aplikacím a způsobům ovládání.[21]

Tabulka 10. Nastavení měniče

Číslo parametru	Název parametru	Rozsah hodnot	Tovární nastavení	Nastavení uživatelem	Komentář
P0010	Volba stavu měniče			1	Aby bylo možné uvést do chodu, musí být parametr P0010=0 (zde lze nahradit parametry přes sériovou linku nebo nastavit tovární nastavení měniče). Pro nastavování některých parametrů musí být tento parametr P0010=1 (pro tyto parametry je v textu u parametru "s"). Po nastavení je třeba opět přenastavit P0010.
P0100*	Volba provozu Evropa/USA		0		hodnota je určena přepínačem DIP dle manuálu
P0300*	Typ motoru			1	volíme asynchronní typ motoru
P0304*	Jmenovité napájecí napětí motoru			230 V	
P0305*	Jmenovitý proud motoru			0,9 A	
P0307*	Jmenovitý výkon motoru			0,12 kW	
P0308*	Účinnost motoru $\cos \varphi$			0	
P0309*	Účinnost motoru			0,7	
P0310*	Jmenovitý kmitočet motoru			50 Hz	
P0311*	Jmenovité otáčky motoru			1350 ot/min	
P0320	Magnetizační proud motoru	0,0 - 99,0 %			pokud není hodnota parametru zadána, vypočte se magnetizační proud při nastavení P0340=1 nebo při ukončení nastavení měniče (P0390=1 nebo 2)
P0335	Způsob chlazení motoru	0 a 1		0	motor má vlastní ventilaci (na hřídeli motoru je umístěn chladicí ventilátor)
P0640	Špičkový proud motoru	0,0 - 400,0 %			že měnit za chodu, hodnota je vztažena ke jmenovitému proudu motoru
P0700	Způsob ovládání měniče	0 - 6	0	6	nastavuje sériovou linku PROFIBUS / Fieldbus (s komunikačním modulem)
P1000	Výběr zdroje žádané hodnoty	0 - 66	2	6	hlavní žádaná hodnota přes komunikační linku PROFIBUSu bez přidavné hodnoty
P1080	Minimální hodnota výstupního kmitočtu	0,0 - 650 Hz	0	0	že měnit za chodu
P1082	Maximální hodnota výstupního kmitočtu	0,0 - 650 Hz	50	60	
P1120	Doba rozběhu motoru	0,00 - 650,00 s	10 s		že měnit za chodu, obsahem parametru je doba nárůstu výstupního kmitočtu z nuly na max. kmitočet po povelu ZAP
P1121	Doba doběhu motoru	0,00 - 650,00 s	10 s		že měnit za chodu, obsahem parametru je doba poklesu výstupního kmitočtu z maximálního kmitočtu na 0,00 Hz po povelu VYP1
P1135	Doba doběhu motoru po povelu VYP3	0,00 - 650,00 s	5 s		že měnit za chodu, obsahem parametru je doba poklesu výstupního kmitočtu z maximálního kmitočtu na 0,00 Hz po povelu VYP3
P1300	Volba módu řízení a regulace	0 až 3	0		Volí se v závislosti napětí na motoru na výstupním kmitočtu. Je nastavena lineární charakteristika U/f=konst., nastavení je určeno pro synchronní a paralelně spojené motory.
P1910	Měření parametrů motoru	0 až 2	0		Parametrem se zapíná automatické měření odporu statorového vinutí. Při nastavení do 1 nebo 2 se po povelu ZAP po dobu cca 20s měří odpor vinutí. Je nutné zrušit povel ZAP a poté jej opět pro standardní provoz zadat.
P3900*	Ukončení nastavení měniče	0 až 2	0	1,2,3	Při nastavení P3900=1,2,3 jsou automaticky vypočteny a přepsány následující parametry: P0344 - hmotnost motoru, P0350 - odpor statorového vinutí, P0346 - doba magnetizace motoru, P0347 - doba demagnetizace motoru, P0611 - tepelná časová konstanta motoru, P2000 - referenční kmitočet, P2002 - referenční proud



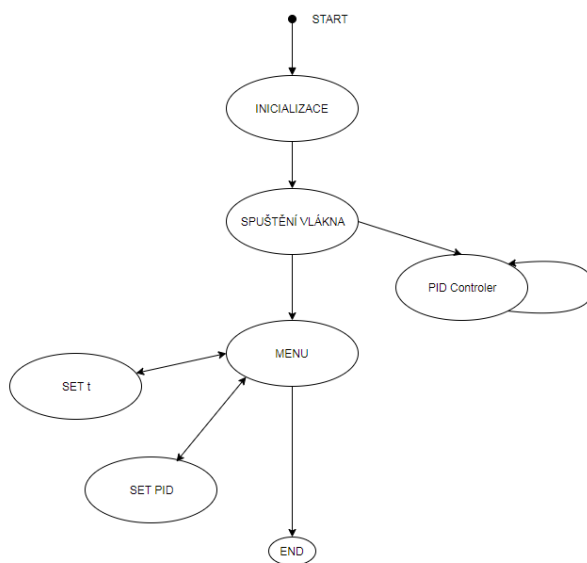
Obrázek 40. Schéma celé soustavy

Po propojení kabely dle Tabulky 8., připojíme síťové napětí. Sběrnice PROFIBUS DP je připojena přes USB rozhraní a dále přes převodník(adapter) PremiumCord USB - USB2.0 na RS485. Je nutno nastavit, nebo zjistit nastavení adresy, které jsou realizovány pomocí binárního přepínače.

Následuje nastavení spojení PROFIBUS DP pomocí GSD souboru, je třeba nastavit rychlost přenosu, bylo nastaveno 9600 bps, Station_type- 1(master), adresy Slave ET200M – č.7 a Mikromasteru 420 – č. 15. atd. Jelikož náš program je poměrně krátký vložil jsem jej celý přímo do knihovny PROFIBUSU DP a přímo pomocí globální proměnné posílal akční zásah na daný výstup A0. jelikož náš algoritmus je v rozsahu 0- 10 a analogový výstup je v rozsahu 15bitů a znaménko, což je 32768dek., převedeme náš výstup regulátoru na tento rozsah($y \cdot 3276$) a ještě přetypovat z float na int. Dále pak posíláme informaci do Mikromastu 420 a to pouze dvoustavově(zapnuto, vypnuto). Aby se nám soustava nerozkmitávala spínám chlazení motoru (obrázek 40.) na hranici $+ 2^{\circ}\text{C}$ nad teplotu zadanou. Tím jsem dosáhl toho, že nám začne motor chladit až při větším rozdílu teplot a soustavu nám nerozkmitává. Nyní očekáváme odpověď slave a jeho hodnotu převedenou z odporového čidla přes odporový dělič a A/D převodník do vstupu AI0 a AI1(jsou k dispozici dvě čidla).

Po rozjetí komunikačního rozhraní PROFIBUS DP byla změřena odezva soustavy(naměřená teplota) a uložena do souboru. Tímto způsobem jsme si ověřili, že regulátor funguje správně. Zde je použita simulace HIL pro kontrolu hardweru celé naší soustavy, při které se ověřuje výkon hardweru a vstupy a výstupy systému. Jako vstup dáme zadanou teplotu, výstupem je pak změřená teplota, která jak je popsáno výše je uložena do souboru. Jedná se o tzv. uzavřenou smyčku, data porovnáme se simulacemi MIL, SIL a PIL.

3.5 Stavový automat



Obrázek 41. Stavový diagram regulátoru

Náš program je jednoduchý stavový automat s vlákem. Ve vlákne nám běží PID regulace, zbytek programu zobrazuje teplotu a nabízí nám možnosti nastavení teploty a PID konstant. Stavový diagram je naznačen na obrázku 41. Při volbě 2. lze nastavit zadanou teplotu obrázek 43. Maximální teplota je kolem 70°C. Při volbě 3. lze nastavovat PID konstanty obrázek 44. Obě menu lze opustit i bez uložení i s uložení parametrů. Volba bez uložení je vhodná například, pokud se chce pouze podívat na nastavení a nechceme tedy nic měnit. Chceme-li regulaci pozastavit využijeme volbu 1. Následně je možno regulaci opět spustit. Pro opuštění programu slouží volba 5.

Tabulka 11. Popis připojení LCD displeje

PIN NO.	SYMBOL	DESCRIPTION
1, 17	3.3V	Power positive (3.3V power input)
2, 4	5V	Power positive (5V power input)
3, 5, 7, 8, 10, 12, 13, 15, 16	NC	NC
6, 9, 14, 20, 25	GND	Ground
11	TP_IRQ	Touch Panel interrupt, low level while the Touch Panel detects touching
18	LCD_RS	Instruction/Data Register selection
19	LCD_SI / TP_SI	SPI data input of LCD/Touch Panel
21	TP_SO	SPI data output of Touch Panel
22	RST	Reset
23	LCD_SCK / TP_SCK	SPI clock of LCD/Touch Panel
24	LCD_CS	LCD chip selection, low active
26	TP_CS	Touch Panel chip selection, low active

Spuštění našeho programu je zachycen na obrázku 42. Jako zobrazovací jednotku jsem použil LCD dotykový display 3,5 palce rozlišení 480x320 pixel. LCD display je připojen pomocí GPIO pinů přímo na Raspberry Pi3. Instalace proběhla pomocí souboru kernel.img na přiloženém cd. Pro změnu parametrů a ovládání kontroléru je nutno použít klávesnici.



Obrázek 42. Raspberry Pi3 po spuštění programu regulátoru.

```
1- SET t: 50.0 C
4- Save and exit
5- Exit
VOLBA:
```

Obrázek 43. Obrazovka nastavení teploty

```
1- P : 4.540
2- I : 0.008
3- D : 32.500
4- Save and exit
5- Exit
VOLBA: █
```

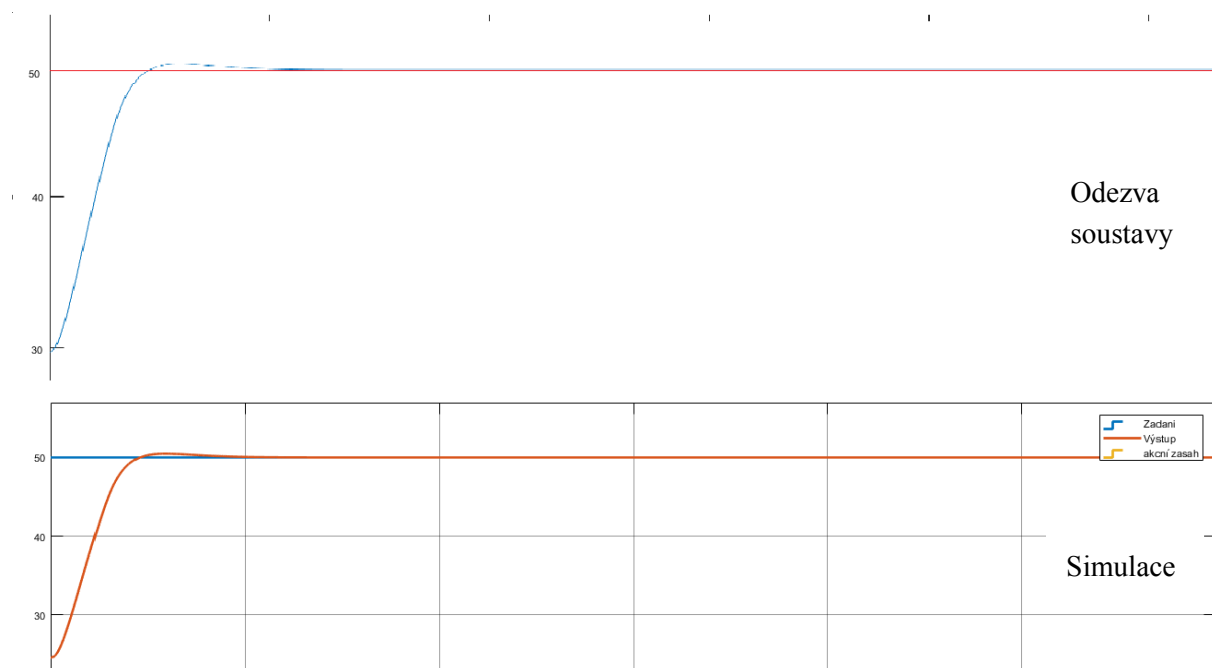
Obrázek 44. Obrazovka nastavení PID konstant

```
Ran
1- STOP
2- SET t
3- SET PID
5- EXIT

Set t: 50.0 C
t: 26.250
```

Obrázek 45. Hlavní obrazovka

Výsledné porovnání odezvy tepelné soustavy a simulace v simulinku je na obrázku 46. Výsledná odezva je podobná a poměrně rychle se nám ustálí.



Obrázek 46. Porovnání odezvy soustavy a simulace v simulinku na zadanou teplotu 50°C

4 Závěr

Účelem této práce bylo projít celý cyklus návrhu regulace, projít jednotlivé fáze. Simulace je velmi silný nástroj jelikož dokáže odhalit chybu dříve než dojde k samotné realizaci. Postupně v nástroji Ident, ladění pomocí funkce tune, použití s-funkce v prostředí simulink a ověření či ladění kódu napsaného například v jazyce C(náš případ). Další možností je generování dat, jež nám pomohou ověřit správnost realizovaného kódu na cílové platformě a následné zpracování dat. Matlab je opravdu mocný nástroj a velmi všestranný.

V teoretické části jsem se zaměřil na popis Raspberry Pi3(naše cílová platforma), teorie simulací SIL, MIL, PIL a HIL, dále teorii PID a PSD regulátorů a v poslední řadě seznámení s PROFIBUS DP sběrnici.

V praktické části je kladen důraz především na simulace v jednotlivých fázích návrhu. Samotný návrh algoritmu může mít mnoho podob. A každý výrobce používá svůj vlastní algoritmus, jak je řečeno i v teoretické části 2.3.1. Už i čistě proporcionální regulátor dosahuje slušných výsledků, zvláště při jednoduché aplikaci jako je naše tepelná soustava.

Pro možnost simulace bylo třeba vytvořit matematický model soustavy v podobě diferenciální rovnice(přenos soustavy), která nám v prostředí Matlab virtuálně nahradí naši fyzickou soustavu. Tímto způsobem můžeme simulovat chování soustavy. Celý proces má obrovské možnosti, jelikož můžeme odzkoušet celý náš algoritmus regulátoru dříve, než je nasazen na konečnou platformu. V případě jakékoli chyby nedojde k poškození soustavy, vše se odehrává ve virtuálním prostředí. Můžeme tedy ihned odezvu na každý náš krok. Po napsání algoritmu PSD regulátoru se mi až na potřetí podařilo soustavu zaregulovat. Paralelní model jsem zvolil z důvodu, že byl pro mě nejlogičtější. Jednotlivé složky jsem mohl upravovat nezávisle na ostatních a tím byl celý návrh jednodušší. Navíc jsem se snažil přiblížit funkčně k regulátoru PSD v simulinku abych mohl celou soustavu snadno naladit a to právě pomocí nástroje tune, což se podařilo. Chování mého regulátoru a regulátoru jak ukazuje obrázek 26. je naprosto stejné a to bylo mým cílem. Stejným způsobem se choval regulátor i po přenesení celého algoritmu na cílovou platformu tedy Raspberry Pi3 a o to nám šlo především.

Jednotlivé fáze návrhu používají různé typy simulací(MIL, PIL...), díky čemuž můžeme kontrolovat správnost návrhu během celého procesu, od návrhu základního algoritmu po výběr periférií, hardwaru, kontrole dostatečného výkonu hardwaru i softwaru, přenosových rozhraní.

Problémy mého návrhu nastaly ve chvíli, kdy jsem chtěl spojit platformu Raspberry Pi3 s ET200M. K připojení jsem měl použít knihovnu napsanou v programovacím jazyce Python dodanou vedoucím práce. Z těchto důvodů jsem celý algoritmus přepsal do jazyku Python a následně jej vložil do této knihovny. Upravil GSD soubor dle potřebných parametrů. Bohužel s časových důvodů se mi nepodařilo soustavu spojit a tak obrázek 46. je pouze demonstrativní. Rozhraní PROFIBUS se používá především pro real-time aplikace. Jeho výhodou je například oproti ethernetu v tom, že odezva je přesně časově definovaná. Například při pohybu různých automatů by nebylo možné bez přesné časové odezvy správně řídit a synchronizovat pohyblivé části stroje. I v našem modelu může dojít k odlišné odezvě pokud se časování bude měnit a dokonce se nám může třeba rozkmitat, což v případě není tak nebezpečné, ale v případě např. ventilů může dojít k poškození a podobně.

5 Použití literatura

- [1] Petr, Pivoňka. *Číslicová a řídicí technika*: Učební text. Brno: Vysoké učení technické v Brně, 2003
- [2] Srovnal, Vilém. *Kybernetika*: Učební text. Ostrava: Vysoká škola báňská – Technická univerzita, 2012
- [3] *Wikipedie* [online]. Wikimedia Foundation [cit. 2018-03-31]. Dostupné z: <https://cs.wikipedia.org>
- [4] UPTON, Eben a Gareth HALFACREE. *Raspberry Pi: uživatelská příručka*. Přeložil Jakub GONER. Brno: Computer Press, 2013. ISBN 978-80-251-4116-8.
- [5] ŠVARC, Ivan, Miloš ŠEDA a Miluše VÍTEČKOVÁ. *Automatické řízení*. Brno: Akademické nakladatelství CERM, 2007. ISBN 978-80-214-3491-2.
- [6] PRUMYSLOVE-PID-REGULATORY-TUTORIAL [online]. Milos Schlegel [cit. 2017-11]. Dostupné z: https://www.researchgate.net/publication/267937755_PRUMYSLOVE_PID_REGULATORY_TUTORIAL
- [8] ÅSTRÖM, Karl J. a Tore. HÄGGLUND. *PID controllers*. 2nd ed. Research Triangle Park, N.C.: International Society for Measurement and Control, c1995. ISBN 1-55617-516-7.
- [9] *Automatizace.hw.cz* [online]. Milos Schlegel, 2010 [cit. 2018-01-10]. Dostupné z: <https://automatizace.hw.cz/zakladni-informace-o-prumyslove-sbornici-profibus>
- [10] [Www.profibus-profinet.cz](http://www.profibus-profinet.cz): *Úvod do PROFIBUSu DP* [online]. 2010 [cit. 2018-03-28]. Dostupné z: <https://www.profibus-profinet.cz/blog/profibus/12-blog/profibus/clanky/15-uvod-do-profibusu-dp-3-dilny-serial>
- [11] [Www.foxon.cz](http://www.foxon.cz): *ÚVOD DO PROFIBUSU DP* [online]. [cit. 2018-03-28]. Dostupné z: <https://www.foxon.cz/cs/blogs/286--dil-z-tridilneho-serialu-uvod-do-profibusu-dp.html>
- [12] VÍTEČKOVÁ, Miluše. *Návrh a seřizování regulátorů přímou syntézou a metodou vnitřního modelu: Controller design and tuning by direct synthesis and IMC methods : teze inaugurační přednášky ke jmenovacímu řízení profesorem pro obor Řízení strojů a procesů*. Ostrava: VŠB - Technická univerzita Ostrava, Fakulta strojní, 2007. ISBN 978-80-248-1432-2.
- [13] BALÁTĚ, Jaroslav. *Automatické řízení*. Praha: BEN - technická literatura, 2003. ISBN 80-7300-020-2.
- [14] *Itnetwork.cz* [online]. [cit. 2018-03-20]. Dostupné z: <https://www.itnetwork.cz/cplusplus/cecko>
- [15] *Raspberry Pi* [online]. [cit. 2018-03-20]. Dostupné z: <https://www.raspberrypi.org/>
- [16] POPP, Manfred. *The new rapid way to PROFIBUS DP: (from DP-VO to DP-V2)*. Karlsruhe: PROFIBUS, c2003.
- [17] *REX Controls* [online]. [cit. 2018-03-31]. Dostupné z: <https://www.rexcontrols.cz/rapid-development-rt-simulace-modelovani>

- [18] *PROFIBUS proces field bus: PROFIBUS Pokyny pro instalaci* [online]. 2013-10 [cit. 2018-03-31]. Dostupné z: https://www.profibus-profinet.cz/images/Dokumenty/PROFIBUS/uvod-k-technologie-PPROFIBUS-cz_FOXON.pdf
- [19] HEROUT, Pavel. *Učebnice jazyka C-2.díl.* 4. vyd. České Budějovice: Kopp, 2008. ISBN 978-80-7232-367-8.
- [20] HEROUT, Pavel. *Učebnice jazyka C.* Praha: [Středisko pro podporu studentů se specifickými potřebami ELSA ČVUT], 2015. ISBN 978-80-7232-383-8.
- [21] Žárovka 2. Popis laboratorní úlohy z předmětu ZSR *Laboratorní model – tepelná soustava se žárovkou 2* VŠB TO.

Seznam obrázků a tabulek

Obrázek 1. Raspberry Pi 3 Model B	2
Obrázek 2. Popis pinů GPIO	3
Obrázek 3. Zpětnovazební řídicí systém s PID regulátorem	5
Obrázek 4.a) paralelní b) sériová realizace PID regulátoru	7
Obrázek 5. Regulační smyčka s respektováním saturace vstupu řízeného procesu	8
Obrázek 6. Klasická struktura PI regulátoru zabraňující unášení integrační složky (anti-reset windup)	8
Obrázek 7. Jiný způsob odstranění unášení integrační složky u PI regulátoru	9
Obrázek 8. Identifikace pomocí T_u a T_n	10
Obrázek 9. Znázornění vrstev podle referenčního modelu OSI	13
Obrázek 10. Aktivní ukončení PROFIBUS RS-485	14
Obrázek 11. Průběh při správném ukončení segmentu	14
Obrázek 12. Průběh při nesprávném ukončení segmentu	14
Obrázek 13. Hierarchie řídicího systému	15
Obrázek 14. Průřez standardním kabelem pro PROFIBUS	15
Obrázek 15. Zařízení zajišťující rozdělení sítě	16
Obrázek 16. Koncept jedné sítě s mnoha segmenty	16
Obrázek 17. Konektory RS485	17
Obrázek 18. Stavový automat PROFIBUS DP Slave	20
Obrázek 19. Ukázka části GSD souboru pro ET200M	22
Obrázek 20. Porovnání změřené odezvy(černá) a výsledného modelu(červená) z Matlabu	26
Obrázek 21. Schéma v simulinku	27
Obrázek 22. Výstup soustavy, akční zásah a zadání	28
Obrázek 23. Blokové schéma pro ladící model v simulinku	28
Obrázek 24. Ladící nástroj Tune Toolbox	29
Obrázek 25. Naladěná soustava PSD regulátoru	29
Obrázek 26. Porovnání průběhů simulací, a) zadaná hodnota(modře)+výstupní teplota(červeně) vstupní data pro simulaci, b) výstup ze simulace PIL, c) výstup ze simulace SIL	30
Obrázek 27. Schéma PID regulátoru	30
Obrázek 28. Pohled na tepelnou soustavu	32
Obrázek 29. Tepelná soustava spolu s žárovko, motorkem a přípravkem pro spojitou regulaci	33

Obrázek 30. Blokové schéma přípravku pro spojitou regulaci	33
Obrázek 31. Zapojení vývodů konektoru CANON 25.....	34
Obrázek 32. Schéma stavebnice Vellman s označením F-KV8003.....	35
Obrázek 33. Schéma převodníku teplotních čidel.....	36
Obrázek 34. Schéma Stabilizovaného zdroje napětí	36
Obrázek 35 3.4.5 Micromaster 420	37
Obrázek 36. Micromastr 420 po odpojení čelního panelu	38
Obrázek 37. Připojení rozšiřujícího modelu pro sběrnici PROFIBUS.....	39
Obrázek 38	40
Obrázek 39 Ovládací panel BOP	40
Obrázek 40. Schéma celé soustavy	41
Obrázek 41. Stavový diagram regulátoru.....	42
Obrázek 42. Raspberry Pi3 po spuštění programu regulátoru.....	43
Obrázek 43. Obrazovka nastavení teploty.....	44
Obrázek 44. Obrazovka nastavení PID konstant.....	44
Obrázek 45. Hlavní obrazovka.....	44
Obrázek 46. Porovnání odezvy soustavy a simulace v simulinku na zadanou teplotu 50°C	45
Tabulka 1 Přiřazení pinů pro konektory Sub-D	17
Tabulka 2 Přiřazení pinů pro konektor M12	17
Tabulka 3. Rozdělení souboru GSD.....	21
Tabulka 4. Znaky a byty rámce hlavičky telegramu PROFIBUS DP	23
Tabulka 5. Znakový rámec v kódu NRZ.....	23
Tabulka 6. Datová rámec hlavičky telegramu(zkratky vis tab. 4).....	23
Tabulka 7. Úvodní oddělovače.....	24
Tabulka 8. Seznam technologických I/O signálů.....	34
Tabulka 9. Nastavení Mikromasteru 420	39
Tabulka 10. Nastavení měniče	40
Tabulka 11. Popis připojení LCD displaye	43

Příloha : CD-ROM disk

Součástí této bakalářské práce je CD-ROM disk, který obsahuje:

- Bakalářskou práci ve formátu PDF
- Programy v C a Pythonu